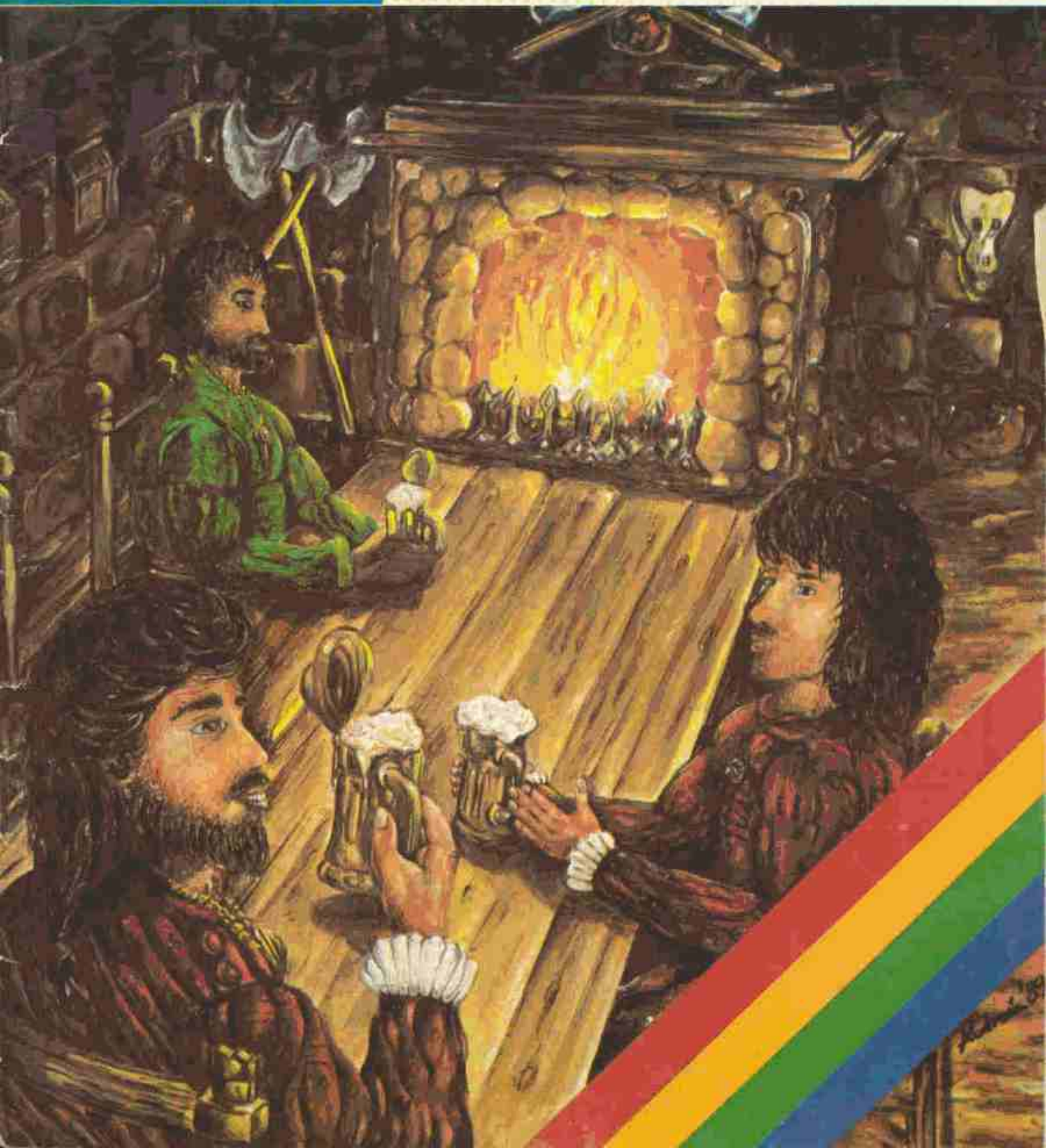


SPECTRUM

VILÁG 22.

+ 4 oldal ENTERPRISE

49 Ft



Az itt látható **S161-S170 Spectrum** 48 kollekciók is megrendelhetők a **PROGRAMKÜLDŐ SZOLGÁLAT** címen keresztül (*Programküldő Szolgálat, Budaörs, Pf.: 12, 2043*). A kollekciók ára: **300,- Ft** (postaköltséggel együtt).

- S161** A/ Winter Games I-II / The Arc of Yesod / Brainstorm / Yabba Dabba Doo / Yie Ai Kung Fu
B/ Zorro / XCEL / Beatin Rider / Devil's Island / Luria Crabs / Yump / Pat the Postman
- S162** A/ Samantha Fox Strip Poker / Action Biker / Run for Gold / Golf 48K / Spookyman / Winlock / BMX Racers / Ostron
B/ Hypocrites Boner Deep / Pimania / Dungeon Master (Creator + Game) / Battle of the Toothpaste Tubes / Bomb Alley / Shuttle Simulator / Shadowfax / Crazy Balloons / 3D Monster Chase / Monsters in Hell / Pakacuda
- S163** A/ Jangler / Colour Clash / Alcatrazz Harry / Nitty Nitty / Super Pipeline 2 / To the Top / Transformers
B/ The Thompson Twins Adventure / Knight's Quest / Worst Thing's Happen At Sala / Dantz / The Prize / On the Run
- S164** A/ Boulder Dash II / Tier / Jasper / Frankie Goes to Hollywood / Sir Lancelot / Viper 3 / Lazy Jones
B/ Invincible Island / Valkyrie 17 / The Inferno / Grand National / Impossible Mission / Pin/Ere
- S165** A/ Invasion of the Body Snatchers / Corridors of Gemon / Black Planet / Confuzion / Boeing 747 Flight Simulator / Give Me Regards To Broad Street / Twin Kingdom Wallay
B/ Tailman / Super Gran / Super Mut / Fireman Fred / Defusion Worms / Two Games / A Day At the Life
- S166** A/ Spinada / Desert Patrol / TAXI / Starquake / Gyroscope / Cauldron / Locomotion / Video Pool
B/ Fairlight / World Series Basketball / Saboteur / Chuckle Egg II / Rumba / Frank Bruno's Boxing
- S167** A/ Buckrogers / The Rocky Horror Show / Tapper / Spy Vs Spy / Spy Hunter / Lode Runner I.
B/ Body / Turmoil / Hunchback 2 / Brian Bloodaxe / Falcon Patrol 2 / Project Future
- S168** A/ Potty Pigeon / World Series Baseball / Starlock / Technician Ted / Moon Cresta / Lode Runner II.
B/ Alan / Formula One / Promania / Merry Xmas / Battle for Midway (PSS) / Monty On the Run
- S169** A/ Subterranean Stryker / Montenegrous / Red Arrows / International Karate / Skyrenger / Frank'n Stein
B/ Jack and the Beanstalk / Driller Tanks / Buggy Blast / Chuckle Egg / Cyclone / Kung Fu / Sabre Wulf / Beaver Ann
- S170** A/ Mugly / Freez Beez / Kokotoni Will / Match Day / B.C. Bill / Underwulfide
B/ Gangsters / Blammy Burgers / Sentinel (Abacus) / Orc Attack / Tutankhamun / Magic Mountain / Borzak / Sorcery

Az "SpV" 20. számában megjelent keresztrefolyó szerencsés nyertesek:

- Csárdás András - Bp. (S150), Csizsar József - Csongrád (S100),
Gedson Andriás - Gyöngyös (S147), Horváth Gábor - Sikső (S05),
Marton Gábor - Nagykanizsa (S80).

Nyerményüket postáztuk!

MINDEN KEDVES MEGRENDELŐNKNEK FELHÍVJUK A FIGYELMÉT, HOGY A TOVÁBBIÁKBAN RENDELESEIKET, REKLAMÁCIÓIKAT A KÖVETKEZŐ CÍMRE POSTÁZZÁK:

PROGRAMKÜLDŐ SZOLGÁLAT

Budaörs, Pf.: 12

2043

PC21

Megnevezés: **BIRTHDAY AND XMAS MEDLEYS**
File-ok száma: 9, Besorolás: Oktató
Memóriafoglaltság (lémezzen): 140K
6 karácsonyi és 6 születésnapi dal megértésével teszi lehetővé szép és ötletes grafikával.

PC22

Megnevezés: **MONGOPOLY**
File-ok száma: 13, Besorolás: Játék
Memóriafoglaltság (lémezzen): 130K
A Ják Ismert MONGOPOLY játék, egy számítógép, egy kulandójáték és egy újraték található a lemezen.

PC23

Megnevezés: **JAPANESE - ENGLISH TUTOR**
File-ok száma: 28, Besorolás: Oktató
Memóriafoglaltság (lémezzen): 230K
Angol-japán nyelvtanító program. Szótáriási gyakorlatok, mondatszerkesztés teszi módosítható.

PC24

Megnevezés: **PC DEMO**
File-ok száma: 58, Besorolás: Utility
Memóriafoglaltság (lémezzen): 250K
Képernyőszerkesztő, demo készítő, demo futtató alkalmazás található a lemezen. Tartólyamokhoz kiváló.

PC25

Megnevezés: **FLASHBAK**
File-ok száma: 4, Besorolás: File kezelő
Memóriafoglaltság (lémezzen): 180K
Közzérő file-kezelő program, melynek segítségével befejezést nyervehetünk a könyvtár le-átstrukturáljába.

PC26

Megnevezés: **BAKERS DOZEN**
File-ok száma: 20, Besorolás: Utility
Memóriafoglaltság (lémezzen): 350K
Szóvagyzerkesztő, naplár információk, disk műveletek, string keresés, összehasonlítás, rendezés stb.

PC27

Megnevezés: **MOVIE DATABASE**
File-ok száma: 51, Besorolás: Adatbázis
Memóriafoglaltság (lémezzen): 230K
20000 film főbb adatait tartalmazza (cím, gyártás éve, rendező, stb.). Moztírajongóknak ajánljuk.

PC28

Megnevezés: **TELESALE V2.2**
File-ok száma: 7, Besorolás: Adatbáziskezelő
Memóriafoglaltság (lémezzen): 210K
Arányváltozó program, kalkulátorral bővíthető. Gyors keresést tesz lehetővé. Kéjparancsoknak ajánljuk!

PC29

Megnevezés: **TERRA TIME**
File-ok száma: 8, Besorolás: Egyéb
Memóriafoglaltság (lémezzen): 150K
Földrajzi ismerőprogram. A világ nagyvárosait, azok földrajzi elhelyezkedését stb. megadja.

PC30

Megnevezés: **BRIDGE PAL**
File-ok száma: 21, Besorolás: Játék
Memóriafoglaltság (lémezzen): 210K
A kártyajáték egy változata. Játshozjuk a gép ellen vagy kettőn egymással. A sebesség változtatható.

PC31

Megnevezés: **PC PROMPT**
File-ok száma: 16, Besorolás: Utility
Memóriafoglaltság (lémezzen): 75K
Speciális memóriaszabvány help-utility. Bonyolultabb parancsok pontos begépeléséhez nagy segítség.

PC32

Megnevezés: **PBASIC**
File-ok száma: 44, Besorolás: Oktató
Memóriafoglaltság (lémezzen): 210K
Alapfokú angol tudást igénylő, BASIC programozásra tanító program. Könnyen kezelhető.

PC33

Megnevezés: **FINGER-PAINT**
File-ok száma: 14, Besorolás: Rajzoló
Memóriafoglaltság (lémezzen): 220K
Rajzoló program, melynek használatához nem szükséges a MOUSE.

PC34

Megnevezés: **CROSSWORD**
File-ok száma: 13, Besorolás: Játék
Memóriafoglaltság (lémezzen): 140K
Keresztrefolyóprogram. Segítségével a lemezen található keresztrefolyók megírfeljelölők.

PC35

Megnevezés: **MINDREADER**
File-ok száma: 14, Besorolás: Utility
Memóriafoglaltság (lémezzen): 220K
Extra funkciókkal rendelkező szövegszerkesztő program. Megáltható lenne a naplár, számdológép stb.

PC36

Megnevezés: **WORLD**
File-ok száma: 4, Besorolás: Adatbázis
Memóriafoglaltság (lémezzen): 130K
Ez egy atlaszútlal és demográfiai adatbázis a világ országairól, népességéről, földrajzról stb.

PC37

Megnevezés: **PACK DISK UTILITIES**
File-ok száma: 6, Besorolás: Utility
Memóriafoglaltság (lémezzen): 450K
Egy 6 funkcióval utility DELDIR, LISTFRAG, NAMEDIR, PACKDISK, TRANSDIR és PARK lehetőségek.

PC38

Megnevezés: **TEST**
File-ok száma: 11, Besorolás: Oktató
Memóriafoglaltság (lémezzen): 110K
Intelligencia-teszt. Segítségével a felhasználó magja is létrehozhat teszt-kereső file-okait.

PC39

Megnevezés: **MINGENY**
File-ok száma: 8, Besorolás: Utility
Memóriafoglaltság (lémezzen): 150K
Képernyő-tervező, szerkesztő program. Látványos szemléltető segédanyag készíthető segítségével.

PC40

Megnevezés: **POKER - BLACK JACK**
File-ok száma: 7, Besorolás: Játék
Memóriafoglaltság (lémezzen): 180K
A lemez a népszerű POKER és BLACK JACK játékokat tartalmazza. Jó grafika, szöveghoztató, játék.

Az itt látható IBM PC XT/AT programlemezek is megrendelhetők (címlapon)

Programküldő

Szolgálat

Budaörs, Pf.: 12, 2043

Egy lemez ára: **300,- Ft** (Postaköltséggel együtt). A továbbiakban az eddig megírdelt PC01-PC20 programlemezük is 300,- Ft-os áron állnak rendelkezésre. A közléseket kérjük, hogy számaingényük a megrendelés-felér együttesüljék jelezze!

Olvásóink riadt hada ragadott tollat a SpV 20. számában közölt „Csillapodó hazai lész” című cikkünkre. Ezekből a levelekből az derült ki, hogy sokan megijedtek: meg fog szünni a SpV. Erről szó sincs: mint ahogy azt már közöltük, a továbbiakban kéthavonta – azaz Idén még négyszer – jelenünk meg. Az alábbiakban az említett cikkekre való olvasói reagálások tengeréből kiválasztottunk két – egymástól teljesen eltérő – levelet, amelyet most közzéteszünk. Mi már eimondtuk a bánatunkat az SpV-ben – most tehát az Olvasóé a szó:

Tisztelt SpV!

Én eddig a SpV. mind a 20 részét megvettem. Olyan tanácsokat szeretnék adni önöknek, amivel növelhetnék a vásárlók, illetve a "kaszettaküldő szolgálat" megrendeléseinek számát. (Ez utóbbi főként teljesen független szolgáltatás – SpV).

1. A tartalom elég jó, a játékokra elég kevés oldal jut. (15 oldal + térkép) Ezt az ismeretlen nyelvek rovására kellene növelni. A Spectrum felhasználóknak 98%-át (Honnan a szám?! – SpV) a Prolog nyelv nem érdekli, a többiek könyvekből tanulja meg a micropologot, számukra elfoglalni 3 oldalt nem érdemes. A hardware ötletek az olvasók 98%-a (Honnan a szám?! – SpV) számára csak „papírpazarlásnak” tűnik, mivel csak a hardware szerelésekhez érdekelnek szólni. Az olyan programok, mint az S.C.S., szintén nem sok embernek jelent valamit, akit pedig érdekli, kis angol tudással is le tudja fordítani az angol szavakat (nincs sok angol szó). (Kedves barátom! Elmondanánk neked, hogy a Spectrum Világ, mint a neve is mutatja, nem "Spectrum játékmagazin", hanem egy átfogó kép a Spectrum és a vele kapcsolatos dolgok világáról – amelybe beletartozik a HW, a Prolog és sok más is. – SpV)

2. A programozástechnika részt hasznos résznek tartom, de a 128K részt nem zeneprogramokra kellene felhasználni, mert akit érdekel, maga is készíthet ilyen programokat. Inkább 128K játék és felhasználói programok (pl. 128K MUSIC Box), valamint programozási ötletek (pl. 48K-s programok átírása 128K-ra) érdekelnek a vásárlókat. Úgy gondolom, a nagyobb memória teszi a 128K-s géppé a 128K-s gépet, ennek felhasználása minden 128K-ot érdekelne. A keresztrefutvány nagyon jó, de kicsit kevés a nyereség (máshol sokkal többet lehet nyerni). A gépi kód tanfolyamban különböző trükköket írjanak le, de ez ne legyen több 2 oldalnál!

(Egy kérdés: láttál már 128K-s gépet működés közben?! – mert figyelembevéve azt a tényt, hogy minden 128K gépen van 48K-üzemmód, a 48K-s programok átírása 128K-ra egyszerűen frentikus ötlet. Főként, hogy az újabban megjelenő 48K-s játékok már intelligensen használják a 128K-s gép 3 csatornás hanggenerátorát. Keresztrefutvány nem lesz több, a nyereségekről pedig annyit, hogy mindenhol annyit a nyereségkeret, amennyit a kiadvány gazdasági vonatkozásai elbírnak. Gépi kód tanfolyam sem lesz a továbbiakban, az pedig, hogy melyik fejezet hány oldal, azt sohasem előre határozzuk meg, mert sokkal nehezebb 2 oldalba belezsúfolni adott információt. Kifér, ahogy kiféri! – SpV)

3. Azokat az örökleteket, amelyek pl. a YOUR SINCLAIR-ben vannak, úgy adják le, hogy egy oldalon 40-50 POKE is legyen. Multiface v. Micro-poker számára használhatóan, pl. UNIVERSAL HERO és a FRIGHTMARE POKE nélkül használhatatlan, és ezekhez POKE-ot egyetlen magyar könyvben sem láttam. Az a legrosszabb, amikor azt feltételezik, hogy az olvasó az "ENTER"-t sem ismeri.

(Szívesen leközöljük a YOUR SINCLAIR poke-jait, akár 40-50-et is egy oldalon, ehhez viszont az szükséges, hogy fizesd elő nekünk E30-ért! Eddig ugyanis saját külföldre és főleg olvasóink által beküldött poke-okra támaszkodtunk. Egyébként mi lesz azokkal, akiknek nincs Multiface-ük?! Továbbá ha poke-ra van szükséged az általad említett játékhöz – és még Multiface-ed is van! –, akkor mi lenne, ha megkeresnéd te magad? Vagy az már nem megy?! – SpV)

4. Én a SpV-ot játékleírásokért és térképekért veszem meg, akkor is megvenném, ha csak ez lenne 32+4 oldalon. Azonban nem mindegy milyenek ezek az információk: hasznosak vagy fölöslegesek (Kinek hasznos, kinek felesleges – SpV). A Killed until dead-hoz adott

információk (régibben STARION) „elpusztítják” a játék értelmeit. Mind a kétfőben a rejtelem megfelejtése a cél és nem a végigjátszás (Nagyszerű észrevétel – viszont mi lesz azokkal, akik nem értenek olyan szinten angolul, hogy erre képesek? – SpV)

Sok humoros részlet hosszadalmassá tesz egy egyébként kevés információt tartalmazó játékleírást. Nincs szükség a "történelmi viselkedésekre" sem. Pl. a NAVY MOVES v. a STALINGRAD esetében a leírás 40%-át is kitépheti egy ilyen "leírás" a program "keletkezési körülményeiről", a software házról, amelyik forgalmazza, "elődeiről" és "utódairól". 3 mondat egy hosszabb leírás esetén elég, rövidebbnél ilyenre nincs szükség. (NEKED nincs szükséged rá! Az SpV viszont új, az olvasmányosság maximális igényével: másoknak van igényük a szellemi táplálásra is, érdekli őket a történelmi háttér vagy az információk a keletkezéséről, stb. Arról nem is beszélve, hogy amikor példának felhozta, nem játékleírások, hanem játékleírások... – SpV)

Sok játéknál ne a főcím screen nyomtassák le, hanem egy jellegzetes pillanatot nyomtassanak. Ennek hiányában kellemetlen csafódások érhetik az olvasót, aki sokkal többre számított. Ha nincs lehetőség erre (pl. rövid leírásokban) szövegesen írják le néhány mondatban, hogy kb. mire számíthatunk (Nem tudom, hogy észrevetted-e, de te szoktuk írni, hogy "mire számíthatunk" – SpV).

A rövid méretű leírások, amelyek a 17. részben még megvoltak, mára már eltűntek. Különösen rossz az, amikor egyszerűen lefordítják a használati utasítást (Melyik volt az? – SpV), még rosszabb, amikor valami jó humoros történetet találunk ki (Hm, olyan már tényleg volt – SpV).

Az ilyen esetekben a célt és a kerettörténetet egy-két mondatban leírhatják, ez fontos is, de hosszabb ne legyen! Screen is csak "hosszabbít", de lényeges információkat nem közöl. Mindig említék meg, hogy milyen a grafika (3D szobák, 3D terem, térkép, kilátunk egy járműből, oldalnézet, felülnézet) (a nyomtatott screen-on, ami szerintem Tényleges információt nem közöl", többek között ez is látható – SpV), és ne azt említék meg, hogy mihez hasonlít (mert lehet, hogy arról a programról az olvasó sosem hallott). A leírásokat lehetőleg úgy írják, ahogy időben történik, és nem összevissza (mint a FOOTBALL MANAGER-nél). Nem tartom szükségesnek a menüt mindig megemlíteni, csak az újrafelhasználható funkciókat. A tömörséget és a pontosságot mindig tartásuk szem előtt. Az a cél, hogy minél több, minél pontosabb leírás jelenjen meg. A 19. rész 2. oldalán lévő leírások elég jól eleget tesznek ennek a követelménynek, ilyenekből lehet 6-8 is. (Egy szerencsénk van: a SPECTRUM VILÁG szerzőit nem úgy hívják, hogy Marx Gábor. Egyébként köszönjük a kiadványnak! Ha annyira nem tetszenek a leírásaink, akkor miért veszed az SpV-t? Egyébként dumálni könnyű: vedd figyelembe azt is, kispajtás, hogy nekünk minden hónapban (illetve most már csak kéthavonta) az egész újságot meg kell írni és teljesen nyomdakészre hozni! Te meg tudnál tölteni 32 oldalt 20-30 tömör játékleírással havonta?! – ezen meglepődnénk. Különösen, ha figyelembe vesszük azt is, hogy mi nem egy iskolában fogjuk a lábunkat napközben, hanem van egy munkahelyünk is, ahol – minő csoda! – olvadják tőlünk, hogy reggel 8-tól délután 5-ig megdolgozzunk a fizetésünkért! – SpV)

5. A hibák, pontatlanságok és a hiányosságok a legrosszabbak. Ebben rekordot volt a WHITE LIGHTNING sorozat, második a VULCAN leírása, ahol több mint 20 hibát találtunk a tesztvéremmel együtt (A tesztvéremet nem mi csináltuk, őt kár volt beleszámolni a hibák közé... Egyébként gratulálunk a nagyszerű eredményhez! Azt nem számoltad meg véletlenül, hogy hány értékes információ volt az átlátható "hibák" mellett?! – SpV)

A hibák:

– fordítási hibák

(Melyek azok?! Konkrétumot ugyanis nem írtál. Ha esetleg a "VULCAN-törést" beszélt, akkor nagyon jól mutatunk rajtad: ugyanis a leírás készítésekor az elérhető összes külföldi irodalmat átbogarászta a SZÁMALK-ban, de csak egyetlen rekordot fedeztünk fel a VULCAN-ról, amely annyi információt tartalmazott róla, hogy megjelent. Következésképpen kis agyunkból kellett megszűrni a leírás minden egyes betűjét!)

– nem mélyedtek el eléggé a játékban (pl. 10. oldalon az esélyeknél)

(MICSODA?!?! A játékkal több, mint fél évig játszottunk (legalább 80 hadművelet)! Milyen alapon állítasz ilyeneket?! A győzelmi esélyek megítélése egyébként szubjektív, a mi tapasztalatainkat tükrözi, tehát nem hibát! – SpV)

– kisebb pontatlanságok (pl. SOUND, MOR és A/M értéke)
(Mivel nem írtad meg, hogy milyen pontatlanságokra gondolsz, erre nem tudunk válaszolni. Egyébként azt is elmesélhetted volna, hogy a SOUND-nak vajon milyen "értéke" van a játékban?)
– SpV)

– olyan dolgok, melyek sosem voltak a játékban (pl. EFF számítása, felderítés elhárítása, 3 pozícióval sosem lép vissza az egység stb.)
(Vak vagy? Szerinted miből számolja az EFF értéket a program – csak úgy, RND-vel? Te nem veszed azt "elhárításként", ha a századot leölvék? Egyébként az, hogy az egység 2 vagy 3 pozíciót lép vissza (1 karakternyi helyet vettün 1 pozíciónak és annyit visszalep!), a lényegesen abszolút nem változtat! Mindent összevetve: az általam KONKRÉTAN felhozott "hibák" mindegyike nevetséges, abszolút lényegtelen, tehát csak kukacokkodás (aki ismeri a VULCAN-t, bizonyára egyetért velünk). A VULCAN-leírás célja – akár a legtöbb SpV-leírásé – az volt, hogy aki odauk hozzá, az azonnal tudjon játszani velem és ne legyen olyan kérdése a játékkal kapcsolatban, amelyre a leírásból nem kap választ. MÁRPEDIG EZ TELJESÜLT!!! A VULCAN az egyik legjobb leírás, amit idáig készítettünk és nem vagyunk hajlandóak eltérni, hogy azt bárki is – teljesen alaptalanul, a levegőbe beszével – kétségbe vonja! – SpV)

Ilyen problémák más játékoknál is előfordulnak SIDEWALK, KNIGHTMARE, CHAOS, GARFIELD c. játékokban (Már megint a levegőbe beszélsz! – SpV). A KING'S KEEP leírás többszörösen használhatatlan.

Ha valamit nem tudnak, írják le, hogy nem tudják, pl. a STORMBRINGER esetében. (Rendben, laírom: nem tudom, hogy miért veszed az SpV-t, ha szerinted annyira rossz! Miért nem veszed mondjuk Mikro Magazint? – SpV)

6. (Itt a levélíró a kezettaklódó szolgálat tevékenységét méltatja az eddigiokben megszokott stílusban. Számos dologban – meglepő módon – egyetértünk vele, tehát igényeit továbbítottuk az illetékes felé. Tartozunk neki még egy megjegyzéssel ezzel kapcsolatban is: ezt a szolgáltatást NEM KÖTELEZŐ igénybe venni! – SpV)

7. 5 évi Spectrum tulajdonosként kb. 100 programról készítettem leírás, kb. 10-ről hosszabbat is (10-20 oldal is lehet), kb. 30 térképet is készítettem. Tudnék Önöknek küldeni ilyeneket, ha ezek érdeklik, válaszoljanak levelükre. Ezzel alaposan megnövelhetnék a SpV színvonalát. (Nagyszerű! Ami idáig az SpV-ben megjelent az sz** (bocs) a vulgáris kitévelért, de a leveledből ezt olvastuk jó), viszont azzal, amit te csináltál, "alaposan megnövelhetnék az SpV színvonalát". Iamered azt a közmondást, hogy "Minden szentnek maga fele hajlik a keze"? Egyébként nyugodtan küldhetsz leírás: ha megfellel az általunk támogatott igényeknek (ami – mint a leveledből is kiderül – elég alacsony), természetesen le is közöljük. Ha nem felel meg, akkor visszaküldjük (meglehet, hogy ugyanolyan módon kommentálva, mint te a VULCAN-t) – SpV)

Minden probléma ellenére is a SpV a LEGJOBB Spectrum szakirodalom (Csak azért, mert az egyetlen ilyen jellegű – SpV). Ez azonban az én magánvéleményem. Közös érdekünk azonban Önöknek is, nekem is, az összes spectrumosnak is, hogy ez a SpV megmaradjon, és színvonala javuljon. Ezért sikeredett ilyen hosszúra ez a levél.

Tisztelettel: MARX GÁBOR (Losztályos gimnazista), Budapest

SpV: az írásodból és a benne foglaltakból akkor is kitaláltuk volna, hogy hány éves vagy, ha nem írtad volna le, hogy hanyadikba jársz. Ez a mentalitás ugyanis pontosan a Te korosztályodra jellemző. Ez a világnézet kezdődik onnan, hogy MINDEN, ami a körülöttem lévő világban létezik, az nem jó (Te persze tudod, hogy hogyan lenne jó!), és végződik ott, hogy MINDEN úgy lenne jó, ahogy azt Te gondold – pedig az CSAK NEKED lenne jó. Esetünkre levettve az úgy hangzik, hogy az SpV idáig nem volt jó, de most jössz Te a hófehér paipán és megmentad ezt a csodátomeget – pedig amiről beszélsz, az csak Marx Gábor Spectrum Világa lenne. Gondolom nyilvánvaló, hogy ezért nem tudunk rád haragudni, annál is inkább, mert mindenkinek ez volt a véleménye az öt körbevevő világ dolgairól, amíg az iskolapadban lögdotta a lábát és nem kellett valamin le is tennie az asztalra (mondjuk pénzt keresni ahhoz, hogy legalább saját magát eltartsa). Utolsó mondataid is azt sugallják, hogy mindezt

építő szándékkal adtad elő. Fiatl barátunk! Amikor majd néhány évvel többet tudsz magad mögött, az Élet majd rádöbbsent arra, hogy hajlandó leszel elismerni (mert el KELL ismerni), sőt, figyelembe venni (mert figyelembe KELL venni) azt a dolgot, hogy ami Neked tetszik, az másnak esetleg nem. Jöhetnek még azzal, hogy az általad kallagnak titulált dolgok nevelő, oktató, ismeretterjesztő és – nem utolsósorban – szórakoztató célszerűek (különös tekintettel arra, hogy a 15 éves, I. osztályos gimnazisták ismeretel ne csak addig terjedjenek, hogy villámgyorran nyomkodják a joystick "lúz" gombját, hanem – teszem azt – tudják, hogy a VULCAN-hadműveletre milyen történelmi szituációban került sor) – de érveink ügyis leperogónak rólad. Most azt mondod magadban, hogy ez süket duma. Az – meg fahányt borsó. Azonban leszel azíves három dolgot figyelembe venni:

1. Nemcsak Te veszed az SpV-t, hanem felnőt családapaik is, akiket a Prolog, a hardware, illetve az általad kirugásra írt egyéb témák érdekelnek – viszont állapozzák a játékleírásokat.

2. Csak két embernek jutott eszébe ebben az országban, hogy a TELJES Spectrum-tábor (és nemcsak Marx Gábor) FOLYAMTOSAN el kellene látni globális információkkal – és ez a két ember nem sajnált erre a célra időzni munkát, fáradságot, időt – és ami a legtöbb: pénzt. Mint a 20. szám bevezetőjéből, a "Sirólaptól" is kiderült, nem gazdasági szempontok vezéreltek bennünket. Mégsem bántuk meg, hogy nem inkább hamburgeresbödét nyitottunk, mert – jól-rosszul, nézőpont kérdése, de – azt csináltuk és azt fogjuk a továbbiakban is csinálni, amit szeretünk! Elismerést, megbecsülést, dicsőletet természetesen nem várunk (főleg nem tőled!) – mindössze annyit, hogy ha valaki levelet ír nekünk, akkor azt normális hangnemben tegye és eddigi munkánkat ne degradálja le teljes mértékben (főleg nem alaptalanul!).

3. A rádió és a tv manapság állandóan azt szajkózza, hogy kis hazánk a lehetőségek országa. Ha ennyi ambíciót érzel magadban, hát tessék: jelentess meg Te is egy Spectrum-újságot! Csinálj meg Te henteset, havonta, kéthavonta 32+4 oldal tömör, de teljes játékleírás! Üzemeltess Te is – természetesen kívánság-műsor alapján összeállított kollekciókkal és 100 forintos áron – kezettaklódó szolgálatot! Teremtsd elő hozzá az eszközöket (a szellemiök ügyis rendelkezésre állnak)! Teremtsd elő az anyagiakat, fizessd ki a nyomdát, hiszen csak 270.000 Ft! Adózz Te is a drága Állam bácsinak a befektetett pénzedért KÉTSZER! Dolgozz és ne csak beszélj a levegőbe! És végül, de nem utolsósorban: nézz szembe Te is az olvasótábor igényeivel!

Elismerjük, hogy az SpV-nek számos hibája van – igaz nem azok, amelyeket te annak vélsz. Mindenesetre az ilyen hideg zuhanyként ható levelek is hozzájárulnak ahhoz, hogy örvénygálatot tartassunk és az újságot szebbé-jobbá tegyük – tehát meghöszöröljük neked is a kritikát.

Tisztelt Szerkesztőség!

Reagálni szeretnék a SpV 20. számában megjelent „Csatlapodó hazai lát” című cikkre. Én 3 éve vagyok egy Spectrum + számítógép tulajdonosa. Ezt a gépet ajándékba kaptam. Véletlenül lettem spectrumos, mert akkor sem nekem, sem az ajándékozónak fogalmuk se volt a számítógépekről és azok különbözőségeiről, bár érdekelt elméletileg a számítástechnika és a számokkal való foglalkozás. Szerencsém volt. Akkortájt elég sok könyv jelent meg a Spectrummal kapcsolatban, amelyekből meguntam a számítógép használatát és Basic nyelvű programozását. Én abszolút amatőr vagyok. Végül nagyon megzerettem a Spectrumot a jó kezelési logikája és a programválaszték bősége miatt, szíveshez állt, megállította lettem, pedig a fent említett akár más típusú géppel is összehozható volna. A felismerhető fejlődés ellenére úgy döntöttem, hogy nem lépek tovább, mert amatőrségem miatt nem biztos, hogy más operációs rendszert is meg tudnék ismerni, úgy mint a spectrumot és a pénzállam sem enged meg 3 évvel ezelőt, amikor a gépet megkaptam, keresni kezdtem az újságokat és a könyveket amelyekben a Spectrummal kapcsolatban írtak. Találtam ilyeneket pld. Orlet, Mikrovilág, Mikro Magazin. Sajnos ezek az újságok fokozatosan csökkentették az információt a Spectrumról és előtérbe kerültek a Commodore témák. Két rossz érzés kezdett eluralkodni rajtam: 1) De jó lenne, ha egy Commodore 64 típusú gépet kaptam volna ajándékba. 2)

A Commodore elnyom minket, persze más típusú gépek tulajdonosait is, a nagyobb üzleti és irodalmi (könyvek, folyóiratok, program) támogatás miatt. Ráadásul az újságokban az olvasók leveleire válaszolva mindig azt írták, hogy azért kevesebb a spectrumos anyag, mert a Spectrum tulajdonosok tústák, gyengék programozásában, nem írnak és ezért nincs elegendő közölhető téma (**Személyeskedt hazugságot! Mesélhetnénk egy-két dolgot lent nevezett újságok szerkesztőségében az időtöltést dító módszerekről, de nincs szükségünk számítógépre – SpV**) Akkor is összehasonlítottuk a Commodore tulajdonosokkal és szapultak minket és más típusú gépek tulajdonosait. Ez bizonyítható meglévő számok fellaporásával. Szerintem nem volt igazuk, mert a külföldön megjelenő lapok mást bizonyítottak a témagazdagságról. (Számuk könyvtárban megtalálható). Ekkor üstökösként megjelent a Spectrum Világ című kiadvány, nagy örömmre.

Az üstökös elnevezés jól tükrözi a kiadvány fejlődését (rohamosan fejlődik) és a „Csillapodó hazai lár” című cikkben leírt példányszám alakulási görbét. Nekem valamennyi szám megvan, még a SpV 7. rész is, de úgy emlékszem, hogy annak a beszerzésével valamilyen problémám volt. Sajnos már elfelejtettem, hogy milyen, mert ha emlékeznek rá, megtudnám mondani a negatív rekord okát. Nagyon tetszik a kiadvány. Szép külső, jó szerkesztés, humoros játékszemeltetők, sok ötlet, programkollektív kiadvány, keresztrejtvény. A SpV 20. rész különösen jól sikerült. Szerencsés a fejléc hasonlósága a SpV 19. rész fejlécével, amely segíti a felismerést, ha az olvasó messziről ránt egy újságospapilonra. Előnyösebb, mint a SpV 19. rész előtti számok címlapjai, melyek szépek, de nem segítik a lap felismerését a többi tarka újság között. Talán ez is számít egy kicsit a példányszám esökkenés szempontjából. Nagyon jó ötlet a játék és egyéb programismeretűk után az ismeretlen nyelvek, a Programozástechnika és a Gépi kód tanfolyam című rovat, mert a hozzárt hasonló amatőrök sokat tudnak tanulni belőle (**Marx Gábor, hallod ezt? – SpV**). Így a kiadvány szórakoztat is, oktat is, igazán hasznos. **(Pontosan ezt a két célt tartjuk a szemünk előtt az egész újságban! – SpV)**

Visszatérve az eredeti témára, üstökösként megjelent a Spectrum Világ című kiadvány nagy örömmre. Végre nekünk is van egy lapunk gondoltam, vagyis nem igaz, amit a régi újságokban írtak, hogy nincs elegendő téma a Spectrumra, hiszen egy többé-kevésbé rendszeresen megjelenő kiadványt is meg lehet tölteni a témával. Az örömmöm nem tartott sokáig, mert egyszer csak megláttam a CoV. 1. részét. Rosszat sejtettem, megint ki fognak szorítani bennünket és a SpV 20. részében már meg is jelent a „Csillapodó hazai lár” című cikk, benne a régi összehasonlításai, több levelet írnak a Commodore 64 típusú géppel rendelkezők, összetartóbbak, gazdagosok a nekik eladott példányszám, míg a spectrumosok 4-5-en vesznek egy példányt, jogsértők és bunkó is van közöttük (**A bunkók mindenhol a lehető legnagyobb számban képviselték magukat, ez nem Spectrum-jellegzetesség – Murphy**) Félreérics ne legyen, nem akarom a Commodore tulajdonosokat bántani, mint Spectrum tulajdonosok, mert elismerem őket és a C-64 tudását, akár köztük is lehetnek, sok ismerősöm van közöttük. De mivel nagyobb a létszámuk, több érdeklődő levelet írnak. Valószínű az ő klubjaikban is lesznek jogsértők és bunkók, majd bizonyítják, mert ezek a tulajdonosok nem a géptípus függvénye. Ők is fognak lényegesen többet készíteni és lapokat kitépni. Csak a nagyobb létszámuk miatt maradnak gazdagosok, talán. A gazdagságosság, amire önként hívatkoznak a mai világban logikailag érthető, de nekünk megmaradt, megszállott Spectrum tulajdonosoknak, mert kisebbség vagyunk és nem gazdagosok vagyunk, jó magyar szokás szerint miért kell csökkentett kiadványszámmal beérni és végül a kiadványról lemondani. Itt egy sajátos magyar problémát kell megoldani. Biztosan van valamilyen megoldás. Sajnos nem vagyok gazdasági szakember, nem tudok hasznos

tanácsot adni. Ha az anyagiakon kívül (ami persze lényeges) Önöknek is érdekük lenne a kiadvány, biztosan nem azt találnák ki, hogy a CoV-t sürítik és a SpV-t csökkentik, míg ki nem pusztul (**Pontosan ezt találtuk ki, a kipusztulás kivételével – SpV**). Ha ez lenne a megoldás, akkor a CoV is egyszer „kipusztul”, ez nem lehet a cél. Máris 2 havonta jelenik meg a SpV című kiadvány, felváltva a CoV-gal, a kipusztulás árnyékával, pedig kiszámítható, hogy a megvásárolt példányszám továbbra is csökkenni fog. Talán ki lehetne használni a Commodore tábor létszámát és érdeklődését és összevonni a 2 kiadványt és akkor nem kellene könnyedén lemondani az egyikről és a gazdagságosság is megmaradna (**Akkor lenne csak a nagy lázadás! A Spectrumosok azért, hogy miért kerül annyiba, amikor csak annyi Spectrum van benne és több Spectrumot kérnek – a Commodore-osok pedig ugyanezt fordítva. Nem tartom jó ötletnek – SpV**). Vagy talán a megmaradt néhány ezer olvasó utca) irratás helyett, névre szólóan, a postai költséggel megnövelt áron postán keresztül kapná meg a kiadványt kikerülve a terjesztés költségeit (Ez ügyben már tettünk lépéseket, ami valószínűleg mindenképp előtt ismert: mindkét kiadvány január 20-ig előfizethető volt az egész évre. Az eddig megjelent, utólagosan megrendelt számoknál pedig azt találtuk ki, hogy a pénzt csak akkor kell befizetni és a középső szalvány hátoldalán lévő „Megjegyzés” rovatban kell feltüntetni, hogy melyik számot rendelni meg az Olvasó. Ezzel a trükkel elkerültük azt, hogy a Postán keresztül ajánlottan és utánvétellel kelljen a megrendelést teljesíteni, amelynek postai díja 33 Ft/50 dkg-nál nem nehezebb csomag. Ez a pénz most nem a Posta zsebébe vándorol (így le kaszálnak rólunk eleget), hanem marad a megrendelőnél. Jó, mi? – SpV). Sajnos a Spectrum Világ olvasói létszáma előre láthatóan csökken különböző okok miatt, az intémeik ellenére is, amelyeket úgyis csak mi olvasunk el, akik vesznek a lapot és a hűtlenséghez nem jut el az intelem, mert nem olvassák a lapot.

Talán a programházak temérdek Spectrumra íródott programja mind gazdasági ráfizetés lenne? (**Nagy részük tényleg az, a sikeres viszont bőven behozzák a dolgot – SpV**) A nagy programválaszték tudatában és a nyugati üzletemberek ismerve, nem hiszem. Ha Önöknek van elegendő közölhető témájuk (játékszemeltetők, programozástechnika, stb.), és ha az anyagiakon kívül (ami persze lényeges) egyéb érdeklődésük lenne a kiadvány folytatásához, kérem kérem megoldást, hogy a SpV elkerülje a kipusztulást. Megállapításaimmal nem kritizálni akartam, csak a Spectrum Világ olvasó Spectrum tulajdonos kisebbség érdekében írtam, mert nem szeretném, hogy elhatárolódjon rajtuk egy érzés, hogy a Commodore elnyomja, még a kedvenc olvasmányától is megfosztja, hátráltatva terjeszkedésével, támogatottságával, létszámukból adódó gazdagságosságával, amelyről ők nem is tehetnek.

U.i.: A Commodore tulajdonosok ugyanolyan emberek, mint mi, Spectrum tulajdonosok, csak szerencsésebbek.

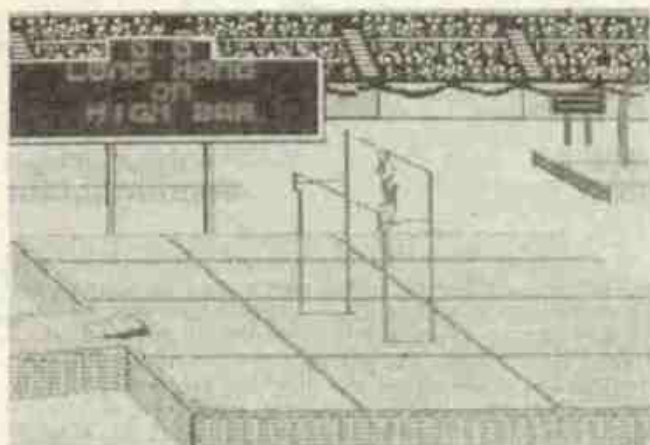
HERR VILMOS, DUNABOGDÁNY

SpV: ezt a levelet nem akarjuk külön kommentálni, hiszen egyrészt magánról beszél, másrészt a tisztázatlan részekre a szövegben már válaszoltunk. Míndenestire figyelemreméltó, hogy a levélben a Spectrum-tábor EGÉSZÉNEK a problémáról esett szó, ellenben az előző levéllel, ahol ezek túlnyomórészt személyes jellegűek voltak – bár gyanítjuk, hogy a levélíró is már jóval túl van a 15. éven. A logikája is rendkívül, valamint legtöbbször öles számmal lát a dolgok MÖGÉ. Köszönjük a megértését és javaslatát! A felvetett problémákkal és javaslatával nagyrészt egyetértünk. Azt mindenestire ígérhetjük, hogy amíg az SpV nem lesz súlyos, 30-40 ezres nagyságrendű ráfizetés, addig az újság MEG FOG JELENNI! – ha másként nem, hát akkor úgy, hogy eltartja a CoV.

Figyelem!

Az SpV segítséget szeretne nyújtani a Spectrum-tábor mindazon tagjainak, akiknek valami olyan információra (leírás, térkép, poke, stb.) van szüksége, amit sehol sem képes megtalálni: a következő számunktól kezdve – egyfajta keres/kinál-rovat keretében – mindenki tetszés szerint hirdethet az SpV hasábjain. A lehetőség – más lapoktól eltérően – **TELJESEN INGYENES!** Minden hirdetést legalább egyszer (de csak egyszer!) le fogunk közölni, de előre felkérjük hirdetni szándékozó Olvasóinkat, hogy a területi szempontokra legyenek figyelemmel! A hirdetéseket pontos névvel és címmel együtt fogjuk közölni, de a hirdetések hitelességét értelemszerűen nem tudjuk garantálni.

GAMES SUMMER EDITION • US Gold



Se szeri, se száma azoknak a számítógépes sport-szimulációknak, amelyek az Epyx cég sikeres pénzbedobós automatáiról készültek és a US Gold forgalmazza őket. Ezeknek a sorát gyarapítja a SUMMER EDITION, amely az eddig már eléggé elkoptatott ötletek mellett jónéhány, játékprogramban még elég ritkán látott sportágat is feldolgoz.

A körítés a sport-szimulációknál már megszokott módon történik: megnyitó ceremónia, az egyes sportágakban eddig fennálló világrekordok listája, érme kiosztása, stb.

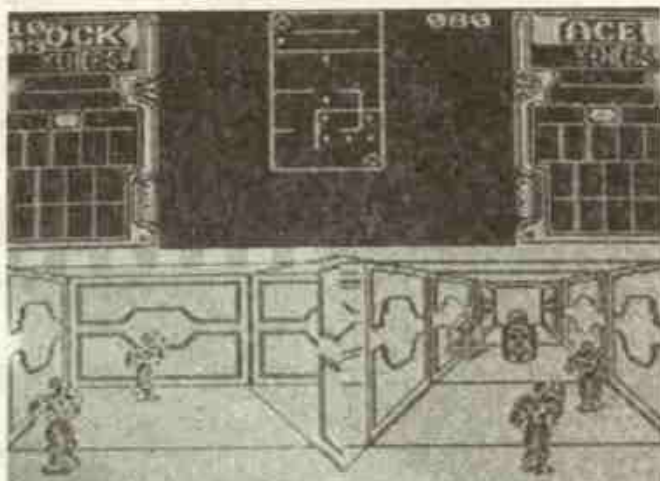
Bejelentkezés után a program érdeklődik, hogy mindegyik sportágban akarunk-e gyakorolni vagy kezdődhet-e az összetett verseny. Számunkra teljesen érthetetlen módon sajnos nincs lehetőség külön egy-egy szám gyakorlására, de gyaníthatóan a kazettás töltés bonyodalmai miatt történt ez így.

A sportágak, amelyekben ringbe szállunk meglehetősen eklektikusan vannak összeválogatva – talán azért, hogy az olyan számok, mint a gátfutás vagy a rúdugrás a joystick kapjon egy kis regenerálódási időt. A következő számokban versenyezhetünk:

- íjászat
- verseny kerékpározás
- toronyugrás
- gátfutás
- rúdugrás
- gyűrű
- felemáskorlát

Akik szeretik a US Gold által forgalmazott sportjátékokat, azok valószínűleg meg lesznek elégedve ezzel is: minden egyes szám szórakoztató játékot kínál, néhány humoros részlettel. A grafika is megfelelőnek minősülhet, viszont a "hangeffektusok" (pontosabban azok a szörnyűségek, amit most annak nevezünk) nem tartoznak a játék legerősebb oldalai közé.

XYBOTS • Domark



Ha már az előbbi játéknál a pénzbedobós automataknál tartottunk, akkor most rögtön folytathatjuk is ugyanott: a XYBOTS ugyanis szintén egy coin-opkonverzió, csak az Epyx/US Gold páros változott Tengen/Domark-ra.

A kerettörténetet a szörnyű és gonosz idegen élőlényekről valamint a két bátor kommandósról mindenki kitalálhatja magának – mi inkább térjünk rá magára a játékra.

A játékban egy vagy két játékos irányít egy kommandópárost egy úrbéli 3D labirintusban. A gátló tényezők szerepét most robotok játsszák, akik huncut módon tüzet nyitnak mindenre, ami eléjük kerül.

A képernyő alsó részén látható az a két nézeti kép, amit a kommandósok látnak. Mivel egymástól teljesen függetlenül irányíthatóak, teljesen más irányban is haladhatnak. A képernyő felső részén látható elkülönítve a játékosokra vonatkozó két felirat, közöttük az aktuális szint térképét találjuk a két kommandós pozíciójával, az akadályokkal, a falakkal és a kijáratokkal a másik szintre.

Mindkét játékos egy adott energiaszinttel kezd a küldetést, amely folyamatosan csökken (különösen ha egy robot orra elé kerülünk). A veszteség a labirintusban elhelyezett extra energiaegységekből pótolható. Ehhez hasonlóan találhatunk még egyéb cuccokat is (pajzs, extra tüzerő, stb.).

Az egymás után a következő szintek – a kategória jó szokása szerint – egyre nehezebbek lesznek: egyre több robot van (amelyek egyre mérgeesebbek is), a labirintusok egyre nagyobbak, néhány helyen teleportokba botlunk, amelyek a labirintusnak egy teljesen más részére teleportálnak bennünket, és így tovább.

A játék igazán akkor a legszórakoztatóbb, amikor két játékos játszik vele – egyedül egy kicsit unalmas a dolog. A grafikára és a zenére sem nagyon panaszkodni: a jó átlagot képviseli mindkettő.

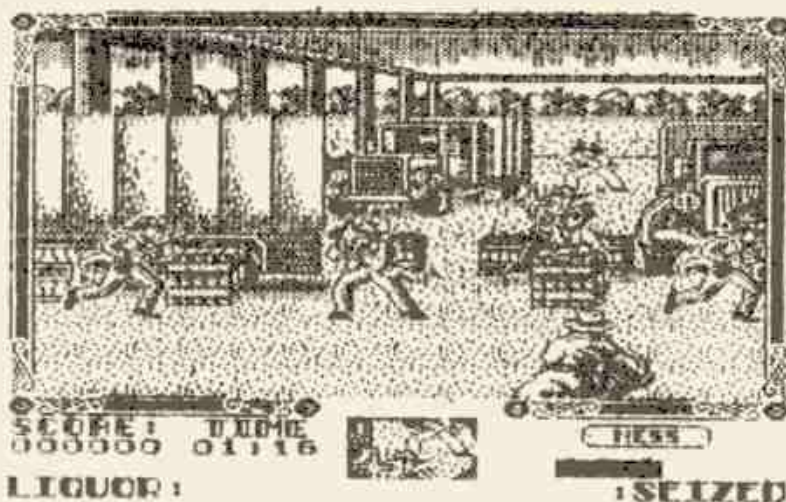
MULTIFACE POKÉ-OK

H.A.T.E. 53246.14 órák energia
Run for Gold 41097.0 órák energia

The Munsters 37891.0 órák idő
Wer Le Mans 26110.34 órák idő

THE UNTOUCHABLES • Ocean

Chicago, 1930. A város Al Capone és a hasonszőrű gengszterek bandáinak terrorja alatt nyög: az utcán farkastörvények uralkodnak, a gátlástalan szeszcsempészek géppisztolyokkal nap mint nap csatatérre változtatják az egykor békés utcákat. A tarthatatlan állapotok mielőbbi rendezése nyilvánvaló megoldást kínál a tehetetlen rendőrség embereinek: a gengszterekkel csak a saját eszközeikkel lehet elbánni. Rövidesen meg is találják a megfelelő embert a feladatra. Az ügynök, Elliot Ness néhány nap múlva a városba érkezik és megkezdí a gengszterek kitarítását...



Talán ezek után az előzmények után elképzelhető, hogy a hasonló című mozifilm alapján készült UNTOUCHABLES számítógépfeldolgozása néhány órányi parázs lövöldözést fog eredményezni a vállalkozó kedvű játékosoknak. A játék pontról-pontra követi a film cselekményét és az alábbi szinteken játszódik:

Áruház (WAREHOUSE): Tippet kapunk egy rendőrségi informatortól egy titkos szeszüzemről és azonnal a helyszínre sietünk, hogy Capone emberei közül minél többet elfogjunk. A képernyőt oldalról látjuk és mindenféle akadályok között kell csetlenünk-botlanunk. Mihelyt elegendő gengsztert elkapunk már is a következő szinten találjuk magunkat.

A híd (THE BRIDGE): Az USA-Kanada határon állunk lesben, ahol információink szerint a tiltott szesz csempészik be az országba. A képernyő előterében hasalunk és a kezünkben szorongatott puskával löhetjük halomra a szeszcsempész urakat. Jobbra-balra gurulva a hídnak újabb részeit is láthatjuk. A képernyő alsó részén lévő kép segít a célpontok biztos eltalálásában: Itt egy távcsőben kinagyítva láthatjuk azt a helyet, ahova tüzelni fogunk.

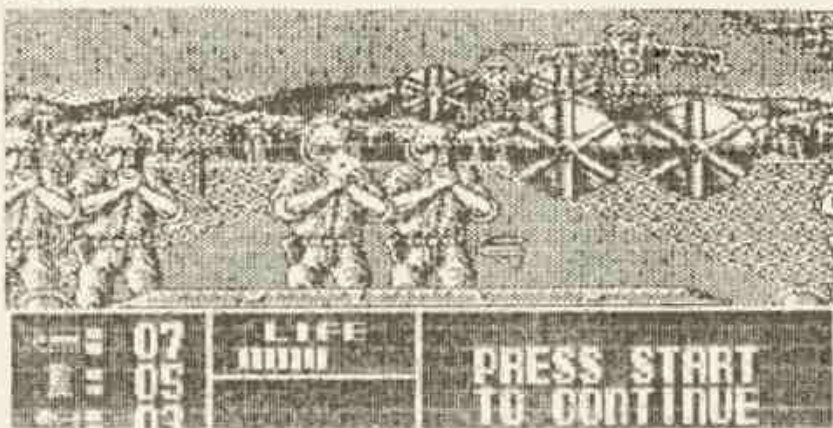
A síkator (THE ALLEYS): A képernyő jobb oldalán állunk gyorstüzelő puskánkkal a kezünkben, előttünk egy kis utcácska. Az utca végében hirtelen autó fékező... Ezen a szinten keresztül kell törnünk magunkat az utcán, hogy kijussunk a vasútállomáshoz, azaz a negyedik szintre. Természetesen ez nem olyan egyszerű, mert az eddig sötét ablakok hirtelen megelevenednek: pisztolyos, ballonkabátos alakok hajóznak ki rajtuk...

...és így folytatódik ez tovább a következő három szinten is, míg egy háztetőn szembe nem kerülünk magával a főbűnőssel, Capone jobb kezével, Frank Nitty-vel.

Az UNTOUCHABLES – azonkívül, hogy szintén film alapján készült – számos közös vonást mutat egy másik nagy Ocean-sikerprogrammal, a ROBOCOP-pal is (nem csodálkoznánk, ha ugyanaz a programozói gárda követte volna el mind a kettőt). Ez a játék is egyszerűen elképesztően jól sikerült! A játékképernyőn ugyan csak két szín van, de a szintér megvalósítása így is nagyszerű. Nagy és precízen animált sprite-ok, szörnyű lövöldözés... mi kell még?! Egy újabb biztos Nr. 1 sikerprogram az Ocean boszorkánkonyhájából: UNTOUCHABLES.

OPERATION THUNDERBOLT • Ocean

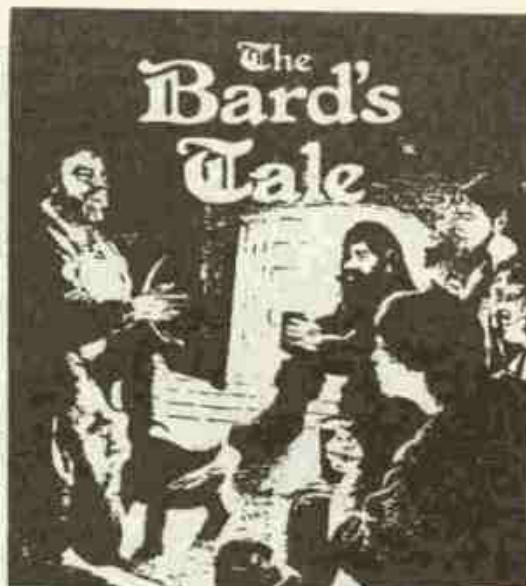
Még alig szorult le a listák éléről az OPERATION WOLF, amikor az Ocean már egy újabb "hadműveletet" zúdított a mit sem sejtő közönség nyakába – ezt most THUNDERBOLT-nak hívják. Nyugodtan lehetne az OW következő nyolc pályája is, mert szinte megtévesztésig hasonlít elődjére (egy újabb csont, amit alaposan le lehet rágni...). A cél természetesen az ártatlanul szenvedő túsok kiszabadítása a nyolcadik szinten, a használati utasítás pedig úgy hangzik: addig is löni mindenre, ami a képernyőn van...



A küldetést néhány tárgy felvételével (golyóálló mellény, plusz lőszer, plusz kézligránát, erősítő ital, elsősegélycsomag, stb.) valamelyest megkönnyíthetjük magunknak – pontosabban nem fognak olyan gyorsan szitává löni bennünket, mint ezek nélkül. Tiszta OPERATION WOLF: 8 szint – ratatatatatatatatatatatata....

Az Electronic Arts software-ház BARD'S TALE-sorozata az egyik legnagyobb sikerű játéksorozat a mikroszámítógépekre forgalmazott adventure-játékok között. Az általa és a hasonló jellegű játékok (BLOODWYCH, CURSE OF THE AZURE BONDS, stb.) képviselő kategóriát pontosabban nem is adventure-nek hívják, hanem Fantasy-kalandjátékoknak vagy még inkább szerepjátékoknak (ROLE PLAYING GAME – RPG vagy FRP rövidítést is használnak). Ezeknek alapjául a Dungeons and Dragons-társasjáték szolgál, amelynek szabályait az amerikai TSR Hobbies dolgozta ki. A D&D-ről akár egy egész könyvet lehetne írni (már több alkalommal meg is történt – angol nyelven), tehát most nem kívánunk részletesen foglalkozni vele. Lényege az, hogy a különböző személyek ("karakterek") szerepébe kell beleélnünk magunkat a Kerd és a Boszorkányság világában – egy vagy több adott feladatot kell megoldanunk, amelyben különféle szörnyek akadályoznak bennünket. Akit esetleg a D&D-nek nemcsak a számítógépes változata érdekel, az hazánkban is hozzájuthat néhány hasonló jellegű könyvhöz: a Rakéta Könyvek-sorozatban már 4 könyv jelent meg Ian Livingstone tollából ("A Tűzhegy varázslója", "Halállabirintus", stb.) illetve az LSI ATSz. is kiadott egy ilyen könyvet magyar szerzőktől ("A sárkány háborúja").

A következőkben a sorozat első részének, "Az ismeretlen meséi"-nek (TALES OF THE UNKNOWN) használati útmutatóját közöljük. A játék meglehetősen terjedelmesre sikeredett, kb. 30-35 pernyi szalagot foglal el. Célja a kijutás a Skara domboldalán fekvő városból, amelynek kapuját azonban varázslattal keltett hőtörésszel zárja el. Cél tehát a gonosz boszorkánymester elpusztítása, amely feloldja a télvárazslatot. A feladatban nemcsak egy embert, hanem egy csapatot irányítunk: a különböző konfliktushelyzetekben mindig más-más karakter tulajdonságaira és képességeire lesz szükségünk – tehát az egyes karaktereinket a lehető legmagasabbra kell fejlesztenünk. A következőkben nem lépésről-lépésre receptet adunk (ez ebben a játékkategóriában szinte lehetetlen lenne, mivel teljesen az adott játékostól függ, hogy milyen karakterekkel harcol, milyen módon hajlja végre a feladatokat, stb.), hanem egy útmutatót, néhány tippet és hasznos tanácsot:



Karakterek tulajdonságai

AC (ARMOUR CLASS): szó szerint "fegyver-osztályt" jelent, de célszerűbb a következő fordítás: sebezhetőség. Minél kisebb, annál jobb. Ennek megfelelően – 10 a legjobb – ezt (vagy az ennél alacsonyabbat) a program L0-val jelzi.

HITS (HIT POINTS): Életerő. Csökkentik a sebesülések, varázslattal viszont gyógyítható. Ebben a sorban szám helyett egyéb jeleket is láthatunk, amelyek a karakterrel történt valamilyen kellemetlenségre utalnak:

DEAD:	Halott
STONE:	Kővé vált
POSS (POSSESSED):	Elmeje fogoly
NUTS:	Elmebeteg
PARA (PARALYZED):	Bénult
PSND (POISONED):	Mérgezett

Ezek közül az állapotok közül megfelelően erős varázslattal magunk is megszüntethetjük a PSND-t és a NUTS-ot, a többi a templomokban 24.000 arany körüli összeg lefizetése ellenében kúrálják.

COND (CONDITION): A HIT POINTS pillanatnyi állása.

SPPT (SPELL POINT): A varázserő nagysága. A varázslás természetesen csökkenti, de visszanyerhető Roscoe energiaközpontjában, vagy napenergiából (az előbbi megoldás pénzbe kerül, az utóbbi lassú).

CL (CLASS): Osztály, foglalkozás. Az alábbi lehetséges osztályokba tartozhatnak karaktereink:

WA (WARRIOR): Harcos. Fő tulajdonsága az erő, fő képessége a harcolás, általában sok a HIT POINTja, bármilyen páncélt és szinte minden fegyvert használhat.

WI (WIZARD): Varázsló. Csak minimum két CLASS CHANGE (lásd később) után érhető el. Az összes többi varázshasználóval együtt fő tulajdonsága az intelligencia, fő képessége a varázslás. Bőrpáncélt viselhet, a többi varázslóhoz hasonlóan kevés fegyvere lehet. Fő varázslattípusa az, hogy szörnyeket idéz meg segítségül.

SO (SORCERER): Illuzionista. Csak CLASS CHANGE után érhető el, páncélt nem viselhet. Varázslatai többsége illúziószörnyeket hív.

MA (MAGICIAN): Mágus. Az egyik alapvető varázsló osztály. Páncélt nem viselhet. Varázslatai többsége támadó jellegű.

CO (CONJURER): Bűvész. A másik alapvarázsló. Páncélja szintén nem lehet, sok gyógyító, védő és világító varázslata van.

RO (ROGUE): Csavargó, tolvaj. Bőrpáncélt viselhet. Fegyverei száma közepes. Képességei a csapatelátárolás és az árnyékban rejtőzés, fő tulajdonsága az ügyesség.

BA (BARD): Bárd, énekes. Viszonylag sok páncélt viselhet, közepes mennyiségű fegyverrel együtt, csak ő használhat hangszereket. Énekével különböző szolgáltatásokat tehet: Pl. világítás, AC javítás, stb.

PA (PALADIN): Lovag. Alig különbözik a harcostól. A legerősebb varázspáncélokot és fegyvereket nem használhatja, de valamivel több a HP-ja.

HU (HUNTER): Vadász. A harcos és a bárd között áll. Nem használhat minden páncélt és fegyvert.

MO (MONK): Szerzetes. Páncélt nem viselhet, fő fegyvere a puszta keze. Szintenként csökken az AC-je.

A karakter képességei 3 és 18 között terjedhetnek, a több a jobb.

The Arc of Yesod • Odin

Az örökélet bavitálát a BE-BE verzió kereszttől fogjuk bemutatni

Tekerjük a magnót a SCREENS végéhez, majd írjuk be

10 CLEAR 84999 FOR a = 65000 TO 65018 READ b POKE a,b NEXT a RANDOMIZE USR 85000

20 DATA 221,33,0,91,17,0,103,62,255,55,205,86,5,67,0,50,230,185,201

RUN (ENTER)

Indítsuk a magnetofont, betöltés után örökösöljük lesz.

ST (STRENGTH): Erő. A harcban van jelentősége: nagyobb erő nagyobb sebzést eredményez az ellenfélen.

IQ (INTELLIGENCE): Intelligencia. Magasabb intelligenciájú varázslók több **SPELL POINT**ot kapnak.

DX (DEXTERITY): ügyesség, gyorsaság. Beszámít a védettségbe.

CN (CONSTITUTION): Aikat, egészség, állóképesség. Magasabb egészség több **HIT POINT**ot eredményez.

LK (LUCK): Szerencse. Káros hatások elkerülésében segít.

LVL (LEVEL): Szint. A karakter életcélja a lehető legmagasabb szint elérése, tapasztalat szükséges hozzá. Természetesen minél magasabb szintű egy karakter, annál jobban használhatóak a képességei és annál legyőzhetősebb.

EXPERIENCE: tapasztalat. Harcban szerezhethjük, a magasabb szint eléréséhez szükséges.

GOLD: arany. Érdekes módon az italon kívül minden pénzbe kerül...

S C M W: A karakter által használható varázslatok fajtája és szintje. A varázsszint nem azonos a karakter szintjével! A varázsló minden második szinten szerezhethet új varázsszintet.

A karakterek fajtái (RACE)

DWARF:	Törpe. Erős, egészséges.
ELF:	Elt, tünde. Viszonylag intelligens.
HUMAN:	Ember. Átlagos.
GNOME:	Gnóm. Nagyon intelligens.
HOBBIT:	Hobbit. Gyors és szerencsés.
HALF-ORC:	Félorc. Erős.
HALF-ELF:	Félelf. Átlagos.

A program kezelése

A játék vezérlése túlnyomórészt menük útján történik, de néhány billentyű is szerepet kapott. Ezek a következők:

- '1-6': a karakterek adatai.
- 'E' (EQUIP): tárgy használatba vétele.
- 'T' (TRADE): tárgy átadása másik karakternek. Ha tárgyat akarunk átadni, a 'T' után a tárgy, majd a karakter számát nyomjuk, ha aranyat, akkor a 'G'-t, a pénzeszeget és a karakter számát.
- 'D' (DROP): tárgy eldobása. Eldobott tárgy kiesik a játékból.
- 'P' (POOL GOLD): a társaság pénzének összegyűjtése.
- 'U' (USE ITEM): tárgy használata. Szükséges a tárgyat használó karakter, valamint a tárgy száma.
- 'P' (PARTY ATTACK): balharc.
- 'I': haladás előre, csak szabad úton.
- 'K': haladás előre, ajtón át is.
- 'J': fordulás balra.
- 'L': fordulás jobbra.
- 'C' (CAST A SPELL): varázslás. Szükséges a varázsló száma, valamint a varázslat neve (Lásd később).
- 'B' (BARD SONG): énekítés. Szükséges a bárd száma, valamint a dal száma (1-6).
- 'BREAK' (vagy két gomb egyszerre): Pause, bármelyik gomb után folytathatjuk a játékot.

Menük

Starhely (GUILD OF ADVENTURERS)

CREATE CHARACTER: Ha van üres hely új karakter számára, akkor aktiválódik. Először fajtát kell választanunk a leendő személyiségnek (vagy visszatérhetünk a főmenühöz), majd megkapjuk X karakter képességeit. Ha megfelelnek, osztályt kell választanunk, majd elneveznünk az egyéniséget, ha nem tetszenek, visszatérhetünk a fajta-menühöz.

REMOVE CHARACTER: Karakter törlése. A törlendő karakter nevét kell beírni.

LOAD PARTY: Társaság betöltése

SAVE PARTY: Társaság kimentése

QUIT GAME: Kiszállás a játékból.

EXIT GUILD: Starhely elhagyása

Felszerelések boltja (GARTH'S EQUIPMENT SHOPPE)

Először az üzletelni kívánó karakter számát kell megnyomnunk (vagy az 'E'-t a bolt elhagyásához), majd a következő menüt kapjuk:

BUY AN ITEM: Tárgy vásárlása. A készletből egyszerre kilenc tárgyat láthatunk. A listát az 'F' billentyűvel előre, a 'B'-vel hátra lapozhatjuk. A vásárlás a 'P'-vel és a tárgy számával történik, más billentyű megnyomására a főmenübe térünk vissza.

SELL AN ITEM: Tárgy eladása. A tárgyaink listáját látjuk, valamint azt, hogy mennyi vételárat kínálnak értük (ez valódi értéküknek a fele). Eladni a tárgy számának megnyomásával lehet.

IDENTIFY AN ITEM: Tárgy azonosítása. A tárgyaink listáján keresztül vizsgálhatjuk, hogy mennyiért azonosítják őket. Azonosításra olyankor van szükség, amikor 7 nevű tárgyat kapunk. Az azonosítás a tárgy árának felébe kerül.

POOL GOLD: Pénz összegyűjtése.

DONE: Vége az üzletelésnek.



Továbbképző hely (REVIEW BOARD)

ADVANCEMENT: Továbbképzés. Ha egy karakter rendelkezik a megfelelő mennyiségű tapasztalati ponttal, akkor megkaphatja a következő szintet.

SPELL ACQUIRING: Varázslatszerzés. Ha varázslónk megfelelő szintű, tetemes summa fejében új varázsszintet kaphat.

CLASS CHANGE: Osztályváltás. A varázslók kiváltsága. Akkor lehetséges, ha már legalább három varázsszintünk van – ekkor átválthatunk másik varázslóosztályba. Célszerű azonban ezt csak a 7. varázsszint elérése után tenni (ez a maximum), mivel visszaváltás az előbbi osztályba nem lehetséges.

EXIT REVIEW: A továbbképző hely elhagyása.

Kocsmák (TAVERNS)

ORDER A DRINK: Ital rendelése. A város délkeleti sarkánál fekszik az egyetlen kocsmá, ahol bort is kaphatunk...

TALK TO THE BARKEEP: Pletykálgatás a kocsmárossal. Kisebbs-nagyobb összegek megnyitják a kocsmáros száját. Érdemes próbákoznival, néhány megszólalásának hasznát is vehetjük.

EXIT TAVERN: A kocsmá elhagyása

Templomok (TEMPLES)

A templomok rendeltetése a gyógyítás. A HP (HIT POINTS) visszaadását 10 arany/pont arányban végzik, komolyabb bajok gyógyítása 24.000 arany körüli összegbe kerül.

Energiaközpont (ENERGY EMPORIUM)

Itt szerezhetjük vissza elhasznált SPELL POINT-jainkat, 15 arany/pont érték ellenében.

Ládák (CHESTS)

OPEN CHEST: Láda kinyitása

DISARM IT: Csapdaeltávolítás fizikai úton

LOCATE TRAPS: Csapdakeresés

TRAP ZAP: Csapdaeltávolítás varázslattal (TRZP)

LEAVE CHEST: Láda elhagyása

Harc

Szörnyekkel való találkozás esetén megtudhatjuk, hogy maga a Halál áll velünk szemben, ilyen és ilyen szörnyek formájában. Ezután dönthetünk, hogy felvesszük-e a harcot, vagy elfutunk, bár ez utóbbi nem mindig lehetséges. Ha harcba kerülnénk, akkor gyakorlatilag a csapatnak az első három tagja harcol, az utolsó három csak mágiával segíti őket a küzdelemben. Harcban minden karakter kap egy menüt, amelyen az alábbi lehetőségek szerepelhetnek:

ATTACK FOES: Támadás. Ha az ellenfelek több csoportban vannak, a csoport betűjét is kéri a gép.

DEFEND: Védekezés.

PARTY ATTACK: Valamelyik saját társ megtámadása.

USE ITEM: A karakternél lévő valamelyik tárgy használata.

CAST A SPELL: Varázslat alkalmazása, csak a varázslóknál. A varázslat kódja szükséges (ld. később) és – esetlegesen – a célpont.

BARD SONG: Éneklés. Csak a bárdnál, az énekek hatását ld. később.

HIDE IN SHADOWS: Árnyékba rejtőzés. Csak a tolvajnál, ha el akarjuk kerülni vele a harcot.

A varázslók által alkalmazható varázslatok:

Néhány tudnivaló a varázslókkal kapcsolatban, mielőtt rátérnénk a varázslataik ködszavainak és azok hatásának ismertetésére:

- A varázslók csak azokat a varázslatokat használhatják, amelyet az osztályuk megenged (azaz amit addig megtanultak) és amelyre elegendő SPELL POINT (SPPT) áll rendelkezésükre.
- Egyes varázslatokhoz speciális célpontok (saját karakter vagy egy ellenség), de vannak olyanok is, amelyek az egész csapatra vonatkoznak vagy egyéb varázslatok (pl. szellemidézés vagy világítás). A különböző szinteken több hasonló jellegű varázslat is előfordul – ilyenkor a magasabb szinten használható hatékonyabb vagy hosszabb ideig tartó, mint az alacsonyabb szinten lévő megfelelője.
- az alapvető varázslók a CONJURER és a MAGICIAN, egy CLASS CHANGE után érhető el a SORCERER, a következőnél pedig a WIZARD.
- Az előbbi pontból kiderült, hogy harmadik – vagy annál nagyobb – szintre lépő varázslót bármikor "átképezhetjük", tehát például a SORCERER-t átalakíthatjuk WIZARD-dá. Ilyenkor a tapasztalati pontjai lenullázódnak, de viszont megőrzi a HP-t, SPPT-t és természetesen mágikus tudását is (azaz az addig már megtanult varázslatokat). A szintváltást lehetőleg csak az adott osztályban a 7. szintet már elért varázslónál csináljunk, mert abba a varázslóosztályba, ahonnan a varázslót "átképeztük" nem lehet visszalepni – ergo az abba az osztályba tartozó magasabb varázslatokat sem tudja megtanulni soha.
- Az a mágus, amelyik a 4 varázslóosztálynak mind a 7 szintjét végigjárta (tehát az összes varázslatot ismeri) ARCHMAGE (Főmágus, vagy – szabadabb fordításban – Hipermultimegamágus) lesz. Ez egy nagyon kellemes karakter.

A varázslók minden varázslata – a varázslat erejétől függően – bizonyos mennyiségű SPPT-fogyasztást eredményeznek. Mint már említettük ez Roscoe Energiaközpontjában pótolható egy jó adag pénzzel vagy nagyon lassan visszanyerhető a napenergiából. A varázslatokhoz szükséges SPPT-mennyiségeket a varázslat szintje mellett, a /jel után tüntettük fel.

Sabre Wulf • Ultimate

Az örökélet bavitelét a BASIC/SCREENS/35936/16/1/2 írástárkappal rendelkező verzióhoz közöljük.

Tekerjük a magnetofont a lökőd elejére, majd írjuk be:

10 CLEAR 24379 LOAD ^CODE LOAD ^CODE LOAD ^CODE LOAD ^CODE LOAD ^CODE LOAD ^CODE POKE 43575,255 PRINT USR 23424

RUN (ENTER)

Indítsuk el a magnetofont, betöltés után örökéletünk lesz.

Conjurer

Szint	Varázslat	Célpont	Hatás
1/2	MAFL	-	Mágikus fény, világításhoz
1/3	ARFI	ellenség	Támadás tüllel (-1-4 HP a célponton)
1/2	SOSH	saját	Láthatatlan pajzs a varázsló védelmére (-1 AC)
1/2	TRZP	-	Csapda eltávolítása varázslattal
2/3	FRFO	-	Fagyasztás, az ellenfelet könnyebb megsebezni, lelassul iránytű
2/3	MACO	-	
2/4	BASK	karakter	Növeli a csapat egy tagjának az ügyességét a harcban
2/4	WOHL	karakter	Valamely karakter gyógyítása 2-8 ponttal
3/5	LERE	-	Mágikus fény, a MAFL-nál sokkal nagyobb hatású
3/4	LEVI	-	Antigravitáció egy ideig. A csapat átrepülhet a csapdák felett vagy a fel/lejáratokon
3/5	WAST	ellenfél	Energiacsapás az ellenfélre (-4-18 HP)
3/5	MAST	ellenfél	Vakítás, az ellenfelek a következő támadásukat elvetik
4/8	INWO	-	Szellemidézés, a csapatban egy hatalmas farkas jelenik meg
4/8	FLRE	karakter	Gyógyítás. 6-24 HP-t viszaszad, de gyógyítja a mérgezést és az őrléket is
4/6	POST	ellenfél	A célpont megmérgezése
5/7	GRRE	-	Mágikus fény, nagy területen (titkos ajtók feltárásához is)
5/7	WROV	karakter	Növeli az egyik harcoló karakter ügyességét
5/7	SHSP	-	Energiacsapás az ellenség csapatára (-8-32 HP)
6/8	MALE	-	Az előbbinél hosszabb ideig tartó antigravitáció
8/9	INOG	-	Szellemidézés, a csapatban egy nagyon erős ogra (mint Apex, a H.O.T.M.-ben) jelenik meg
7/12	FLAN	-	Gyógyítás. Ugyanaz mint a FLRE, de a csapat mindegyik karakterére vonatkozik
7/15	APAR	-	Teleportálás dungeon-on belül.

Magician

Szint	Varázslat	Célpont	Hatás
1/3	AIAR	-	Légpajzs a varázsló védelmére (-1 AC)
1/3	VOPL	karakter	Mágikus erőt ad a karakter fegyverének a harcban, nagyobb sebet okoz az ellenfélnek
1/2	STLI	-	A csapat körül lévő fémtárgyak csillogtatása, ezaz mágikus fény rövid ideig
1/2	SCSI	-	Helymeghatározás a dungeon-on belül
2/5	WIST	ellenfél	Energiacsapás, a célpont "megöregszik", ezaz gyengül a támadása és védekezése
2/5	MAGA	karakter	Ugyanaz, mint a VOPL, de jóval hatékonyabb
2/4	HOWA	ellenfél	Energiacsapás egy ellenséges élőhalott ellen, -6-24 HP erejéig
2/5	AREN	-	Csapdafigyelő
3/8	MYSH	-	Energiapajzs a csapat körül, -2AC minden karakteren
3/6	OGST	karakter	Árny energiát ad egy karakternek, mintha az ogra lenne
3/7	MIMI	-	Növeli az egész csapat fegyvereinek hatékonyságát (-2AC mindenkin)
3/8	STFL	-	Energiacsapás tüllel az ellenséges csapat ellen, mindegyiken -6-24 HP
4/8	SPTO	ellenfél	Energiaelvonás az egyik ellenféltől, -12-48 HP erejéig
4/7	DRBR	-	Sárkányláng. Hasonló az STFL-hez, de minden ellenfélre 8-32 HP-t sebez
4/7	STSI	-	Mágikus fény, nagy hatással (titkos ajtók feltárásához)
5/8	ANMA	-	Antimágia a csapat védelmére (pl. sárkányláng vagy illúziószörnyek ellen)
5/8	ANSW	-	Mágikus kard (JOE THE SWORD) segítségül hívása a csapat védelmére
5/8	STTO	ellenfél	Kővé dermedt egy ellenfelet (megelőzhető kőszobrot halott kőszoborrá)
6/9	PHDO	-	Ideiglenes ajtó. A következő lépésre felemeli a falat a csapat előtt
6/10	YMCA	-	Mint az AIAR, de az egész csapatnak és végtelen ideig (-2AC mindenkin)
7/12	REST	-	Az egész csapat HP-ja a maximumra emelkedik.
7/14	DEST	ellenség	Halálcsapás. Egy ellenség kivégzése.

Sorcerer

Szint	Varázslat	Célpont	Hatás
1/3	MJJA	ellenfél	Energiacsapás, -2-8 HP a célponton
1/2	PHBL	-	Illúziópajzs. A csapat tagjait nehezebben látja az ellenfél (-1 AC mindenkin)
1/2	LOTR	-	Csapdakeresés
1/3	HYIM	-	Hipnózis. Az ellenséges csapat összes tagja elveszi a következő támadást
2/4	DISB	-	Védekezés egy illúzióvarázslat ellen. A csapat tagjai nem hisznek az illúziószörnyben
2/4	TADU	-	Dummy segítségül hívása
2/4	MIFI	ellenség	Energiacsapás. Akár a MJJA, de -3-12 HP folyamatosan a szintváltásig
2/4	FEAR	-	Félelem. Csökkenti az ellenséges csapat védekezésének és támadásának a hatékonyságát
3/5	WMO	-	Illúziófarkas segítségül hívása (addig harcol míg legyőzik vagy DISB-et nem alkalmaznak ellene)
3/6	VANI	-	A varázsló láthatatlanná válik az ellenség számára (-5AC). Hatékonyan lehet vele támadni
3/6	SESI	-	Hatékony csapdakeresés
3/5	CURS	-	Átok. Az ellenfél alig tud valami kárt tenni a csapatban
4/7	WNA	-	Illúziósoldos segítségül hívása
4/6	INVI	-	Láthatatlanság a csapatnak (minden karakterünkön -4 AC)
4/7	CAEY	-	Teljes sötétség a hadszíntéren (tolvajok előnyben)

MULTIFACE POKE-OK

Sanxion	35028,0	nincs idegen	Stormlord	34092,0	sérthetelenség
Last Crusade	43076,0	örökélet	Op. Wolf 128	39728,0	nincs ellenség-tűz

Szint	Varázslat	Célpont	Hatás
5/8	DIIL	-	Akár a DISB, de az összes illúziószörnyet megsemmisíti (a sajátjainkat is!)
5/7	WIOG	-	Illúzióogre segítségével hívása
5/8	MIBL	-	Energiacsapás tűzzel. Akár a MIJA és MIFI, de az összes ellenfélén és -10-40 HP
6/10	WIDR	-	Illúziósárkány segítségével hívása
7/12	WIGI	-	Illúzióóriás segítségével hívása
7/11	SOSI	-	Az összes csapda felfedezése végtelen ideig.

Wizard

Szint	Varázslat	Célpont	Hatás
1/6	SUDE	-	Szellemidézés, egy zombi vagy egy halott harcol a csapatunkkal együtt
1/4	REDE	-	Energiacsapás, -16-80 HP az élőhalott ellenfeleknek
2/8	LESU	-	Szellemidézés, kisebb démon segítségével hívása
2/8	DEBA	ellenfél	Energiacsapás, -32-128 HP az ellenség oldalán harcoló démonnak
3/10	SUPH	-	Szellemidézés, egy közepes erejű ghoul segítségével hívása
3/10	DISP	karakter	Gyógyítás a csapat valamely megmérgezett karakterén
4/12	PRSU	-	Szellemidézés, közepes erejű démon segítségével hívása
4/11	ANDE	karakter	Egy halott karakter feltámasztása a harcra
5/41	SPBI	ellenség	Egy ellenség átállítása a mi oldalunkra
5/14	DMST	-	Ugyanaz, mint a DEBA, de a démonok csapatára
6/15	SPSP	-	Szellemidézés, élőhalott (spectre) segítségével hívása
6/18	BEDE	karakter	Hasonló az ANDE-hez, de a feltámasztott karakter életben is marad 1 HP-vel
7/22	GRSU	-	Szellemidézés, nagy erejű démon segítségével hívása.

A bárdok dalai

A bárdok énekelésével nem az ellenséget tudjuk hetedhét határba riasztani (bár az sem elképzelhetetlen...), hanem a varázslók egyes trükkjeihez hasonló gyógyító (vissza néhány HP) és energiát adó (-AC pontok) hatásokat érhetünk el vele. Egy bárd mindig csak egy dalt énekelhet, ha egy másikat határozunk meg neki, akkor az eddigit abbahagyja (megszűnik a hatása is) és az újat kezdi el. A daloknak az alábbi hatása van:

Dal	Hatás
1. FALKENTYNE'S FURY	Harci dal, növeli a csapat támadásának hatékonyságát
2. SEEKER'S BALLAD	Tungaram-nóta, világít, amikor bókászunk, de növeli a támadást is fegyveres harcban
3. WAYLAND'S WATCH	Nyugtató, a csapat kevesebb sérülést szenved harcban
4. BADHR'S KILNFEST	A "Kalmopyrin helyett" c. kompozíció. A bárdot gyógyítja mázskálás közben, a csapat sebeit harc közben
5. TRAVELLER'S TUNE	Gyorsabbá teszi a harcolókat, nehezebb lesz őket általáni
6. LUCKLARAN	Antimágia. Védettebbé teszi a csapatot a különböző varázslatok ellen.

A következő számunkban folytatjuk a BARD'S TALE leírását a feladat ismertetésével, néhány hasznos tippel és jótanáccsal, valamint az összes helyszín részletes térképével.

Bruce Lee • Datasoft / US Gold

Az örökélet bevételét a BASIC/6912/35690 file-térképpel rendelkező programhoz ajánljuk.

Tekerjük a magnetofont a főkérdő elé, majd írjuk be:

10 CLEAR 24999

20 FOR a = 25000 TO 25023: READ b: POKE a,b: NEXT a: NEW

30 DATA 221,33,192,98,17,96,139,62,255,55,205,86,5,62,0,50,83,202,50,91,202,195,0,226

RUN (ENTER), a COPYRIGHT szöveg megjelenése után írjuk be:

RANDOMIZE USR 25000 (ENTER), majd indítsuk a magnót, és betöltés után örökéletünk lesz.

Underwurlde • Ultimate

Az örökélet bevételét a BASIC/6912/34820 file-térképpel rendelkező verzióhoz közöljük.

Tekerjük a magnetofont a főkérdő elé, majd írjuk be:

10 CLEAR 26000: LOAD "CODE": POKE 59376,0: POKE 59377,0

20 FOR a = 45050 TO 45100: POKE a,255: NEXT a

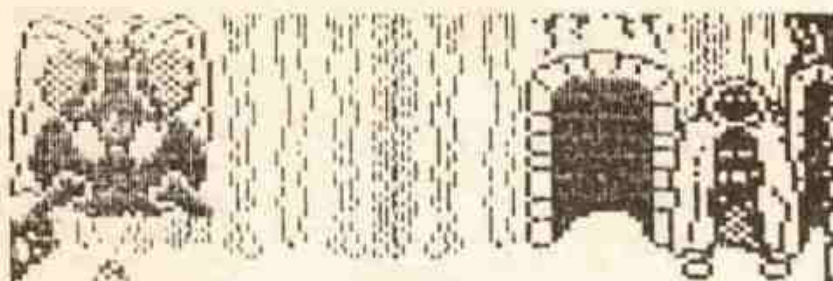
30 RANDOMIZE USR 26610

RUN (ENTER)

Indítsuk a magnetofont, és betöltés után örökéletünk lesz, mindazon túl ellenségeink sem lesznek.

MULTIFACE POKE-OK

Artura	32138,0	örök energia	Frontline	62499,0	örökélet
Mad Mix	39932,x	x = életek száma	Oriental Hero	28995,12	örökélet

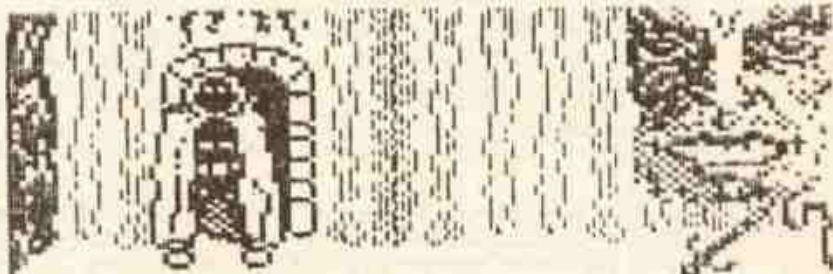


Ha minden igaz, akkor legutóbb a kulcsok ismertetésénél hagytuk abba a HEAVY ON THE MAGICK nevet viselő szörnyűség leírását. Ha már úgyszólván tárgyakról van szó, akkor rögtön folytathatjuk is a következő csoportjukkal, nevezetesen a talizmánokkal. A leírás elején már futólag volt szó az INVOKE varázslatról, amit a Grimoire varázskönyv felvétele után használhatunk – szellemidézésre. Négy szellem áll a rendelkezésünkre (ASTAROT, MAGOT, BELEZBAR és ASMODEE). Mindegyik különböző szolgáltatásokat nyújt, megidézésük az említett igével valamit a kívánt szellem nevének

begépelésével lehetséges, pl. INVOKE BELEZBAR. A szoba velünk ellentétes sarkában füstgomoly keletkezik és a program közli, hogy "Ödémónsága" színe előtt vagyunk... majd azt, hogy nem vagyunk védve ellene (YOU ARE NOT PROTECTED) – és a következő pillanatban már a purgatóriumban (FURNACE) találjuk magunkat, ahol a tisztító tű lángja szépen elemésztí a derék Axilt. Szóval valami védelemre van szükségünk – Apex-nál érdeklődve meg is tudhatjuk, hogy egy-egy talizmánt kell előbb letennünk. Mielőtt sorban átnéznénk a talizmánokat, néhány megszívlelendő jótanács démonidézéssel kapcsolatban:

- a démon mindig a talizmánja FELETT fog megjelenni, tehát mielőtt megidézünk, át kell mennünk a szoba másik oldalára – különben megjelenésekor elveszítjük az összes életerőpontunkat (vagyis kaputt).
- a démonnal gyorsan közölnünk kell a kívánságunkat, mert ha nem szólunk hozzájuk, egy idő után közlik velünk, hogy csak az idejüket raboljuk – és irány a FURNACE-szolárium. Ha félregépeltek volna a parancsot, akkor baktassunk át a szomszéd szobába (ilyenkor eltűnik), majd a visszatérés után idézzük meg újra a demont, mert az újbóli begépelésre már valószínűleg nem lesz elég idő.
- a démonokkal való kommunikáció ugyanúgy zajlik és ugyanazok a szabályok vonatkoznak rájuk, mint mondjuk Apex-nél: először a nevüket kell begépelnünk (természetesen a " után), majd a vessző után azt, amit akarunk tőlük (pl. "ASTAROT, WOLFDORP vagy "MAGOT, GRIMOIRE). * <név>, THANKS begépelésével természetesen őket is elküldhetjük.
- soha ne idézzünk meg demont, amíg rajtuk kívül egy másik élőlény is tartózkodik a helyszínen, mert szellemi túlterhelés következtében el fogunk halálozni.
- a talizmán mindig ott marad ahol letettük – tehát ha mondjuk ASTAROT megutaztat bennünket, akkor az újabb teleportáláshoz előbb vissza kell menni a talizmánjához.

SWORD (Kard): ASTAROT démon elleni talizmán. WOLFDORP-ban van (132). ASTAROT a helyi BKV-nak felel meg: jegyeket, bérleteket ugyan nem kér, csak a hely nevét, ahová menni akarunk. Az előző számban a térképénél már megjegyeztük, hogy csak olyan helyszínre lehet teleportálni, amelynek a neve egy szóból áll (kivéve a SOTHIC COMPLEX – itt SOTHIC-kal kell rá hivatkoznunk). A teleportálás minden helyszínénél ugyanabba a szobába történik. A helyváltás ezen módja ugyan elég praktikus, viszont újbóli alkalmazásához előbb vissza kell mennünk a helyszínre, ahonnan az utolsó teleportálás történt. Néhány tárgyat varázslat véd attól, hogy eredeti helyéről elhozzuk – ilyenkor még jobban jön a teleportálási lehetőség. (ASTAROT egyébként már ismerős lehet a Gargoyle-rajongóknak: a sprite a TIR NA NOG-ból lett átvéve.)



SUNFLOWER (Napraforgó): SOTHIC COMPLEX-ben találjuk (287), egy ajtó mögött. Egyrészt 2-vel növeli tapasztalati pontjainkat, másrészt megvizsgálásakor kiderül, hogy – ha jól fordítjuk – az van ráírva: "négyen teaszünetet tartanak" (FOUR FOR TEA FREE). Ez talán utalás a négy démonra – a tárgy mindenestre egy talizmán, amely MAGOT ellen nyújt védelmet a megidézésekor. MAGOT-ot megidézve a kért tárgy leihelyét kaphatjuk meg tőle (szektor szerint). FONTOS MEGJEGYZÉS! Az előző számban megjelent térképen és a leírásban mi nem azt a számozást használtuk, mint a program, mert ezt körülményesnek tartjuk. A MAGOT által adott szektorszám a következőképpen értendő: az első szám a szint száma, a második kettő pedig a szoba sorszáma (balról jobbra, alulról felfelé számolva). Ezzel a módszerrel pl. a startszoba koordinátája 220, a FURNACE pedig 184.

MANTIS (Tücsök): GORBURG-ban található (312). Sajnos a szobához egy Wyvern-t is mellékeltek, ami az istennek sem (pláne nem egy Axil-nak) hajlandó onnan eltakarodni – rajta tehát némi robbantást kell végeznünk. Miután távozott a másvilágra, vizsgáljuk meg a mögötte lévő tárgyat: egy 20-as szám van belevéve. Fogalmunk sincs, hogy ez mit jelenthet – mindenestre a tárgy a BELEZBAR megidézésekor védelmet nyújtó talizmán. Figyelem, őt csak az idézheti meg baj nélkül, aki már legalább ZELATOR (2^o–9^o) szintre eljutott! BELEZBAR (szintén valami izeltfábú) valószínűleg MAGOT konkurrenciája: a kért tárgy leírását kaphatjuk meg tőle.

A negyedik démonnak csak a nevét mondjuk meg: ASMODEE, a Démonok Királya. Csak a filozófus szintre eljutott Axil tudja baj nélkül megjeleníteni. Hogy milyen talizmán szolgál ellene és mire használható? Ez titok – mindenki kipróbálhatja maga... (hehe!)

A tárgyak további csoportját képezik az élelméként szolgáló dolgok, amelyek felvételük után – hangoz csámcsogás (YUM YUM) közepette – rövid úton Axil bendőjébe vándorolnak. Főhősünk közli, hogy ez finom volt (IT'S TASTY) és növekszik néhány ponttal az életerőnk. Élelméként szolgálnak:

- a korsók (JAR), több is van belőlük. Vigyázat, amelyek csak úgy néz ki, mint a korsó (LOOKS LIKE A JAR), az a bürök (HEMLOCK) és 4 ponttal csökkentli az energiánkat!
- a torta (CAKE), ami úgy néz ki mint egy szikla.
- a WRAITHWALE-ben (272) található cipő (LOAF). Ez 7 ponttal növeli az energiánkat.
- azok a csontok (BONE), amelyek hússal vannak borítva (SHEARED WITH MEAT). Csontból van vagy 10 darab, amelyek közül néhány semmire sem használható, néhány kaja, néhányra pedig később lesz szükségünk.

The New Zealand Story • Ocean

A címképrnyő megjelenésekor billentyűzzük be: FLUFFY, akkor meg fog jelenni a képernyőn a CHEAT felirat, s örökéletet kapunk a további játékhoz.

Mivel az ételként szolgáló tárgyak az energia növelésére szolgálnak, természetesen megvan az ellentétük is: egyes tárgyak funkciója ugyanis kizárólag az energiánk csökkentése. Általában ilyenek azok a tárgyak, amelyek csak hasonlítanak egy másik tárgyra, valójában azonban meg vannak mérgezve, pl. egy pénzszacskó, egy korsó, stb. Vannak viszont olyan tárgyak is, amelyek vizsgálatánál nem az derül ki, hogy hasonlítanak valamire – mégis mérgezőek. Ezek az alábbiak:

SNAKE (kígyó): több is van belőlük, ha felvesszük valamelyiket, az jól meg fogja marni szegény Axil-t (úgy kell neki, miért vette fel?!). Ugyancsak kígyók azok a tárgyak is, amelyek úgy néznek ki, mint egy szikla (**LOOKS LIKE A ROCK**). Az a tárgy viszont, amelyik úgy néz ki, mint egy kígyó, egy szerencsét hozó tárgy.

HEAD (fej): kettő van belőle, mindkettő **METHOS**-ban. Az egyik csak úgy néz ki (ez az **ULNA**, amire később szükségünk lesz), a másik meg a -7 energiapont nevű dolog.

BOOK (könyv): ebből is kettő van. Már találkoztunk velük a startszobában.

Az eddig ismert tárgyak közül is volt már kabala, azaz olyan, amelyek – egyéb funkciójuk mellett – plusz **LUCK**-pontokat eredményeztek, de vannak olyanok is, amelyek csak arra szolgálnak, hogy a szerencsénket növeljék. Szerencsére nagy szükségünk van, mert ettől függ, hogy egy robbantással mennyi energiapontot tudunk az ellenfelünkön sebezni (ha a **LUCK** 7 vagy 8, akkor már nyugodtan felvehetjük a harcot robbantással azon ellenfelek ellen is, amelyek **Apex**-hez hasonló erejűek). Csak kabalaként szolgáló tárgyak az alábbiak:

FOOT (láb): **WOLFDORP**-ban van (133), talán valamelyik erre járó Werewolf futóműve lehetett. A farkasláb általában minden kalandjátékban nagy szerencsét hoz a tulajdonosának (ellentétben azzal a farkassal, amelyik azelőtt használta) – itt +4 **LUCK**-pontot jelent nekünk. Célszerű mindig ezt a kabalát magunknál tartani, ez hozza a legtöbb szerencsét.

IRON CLASP (vaacsat): **WRAITHWALE**-ben akad az utunkba (251). Úgy néz ki, mint egy kígyó – de ha felvesszük kiderül róla, hogy csak egy csat, amire egy sellő van rávésve. +1 **LUCK**.

BALL (labda): **SLYMOLE**-ban találjuk meg (217), vizsgálata után kiderül, hogy rézből van. +1 **LUCK**.



A tárgyak következő csoportja a nagyobb ellenfelek elleni védelemre szolgál. Ezeknek az ellenfeleknek a felbukkanása mindig csak egy bizonyos helyszínen (vagy annak szobájában) várható. Mivel életerő és támadási pontok tekintetében **Apex**-hez hasonló kaliberűek, robbantással nem sokra megyünk ellenük, teljesen esélytelenek vagyunk a harcban (bár nagyon nagy szerencsével és tapasztalattal esetleg robbantással is legyőzhetőek) – ezért mindegyik ellen egy-egy bizonyos tárgy birtoklásával kell védekeznünk. Az adott el-

lenségek ekkor is megtámadnak bennünket, de mielőtt hozzánk érnek, azonnal elpusztulnak. Nézzük sorban a védelmünkre szolgáló tárgyakat:

NUGGET (róg): **METHOS**-ban található egy asztalon (474). Megvizsgálása után kiderül róla, hogy ezüstből van – nagyszerű, vegyük fel és vigyük... azaz csak vinnék: ugyanis mielőtt felvesszük, valami rondaság emelkedik a levegőbe a ládából és nem enged ki bennünket. Ezén nem segít semmitféle beszéd és bűbáj, szóval – ha el akarjuk vinni – más megoldáshoz kell folyamodnunk. Ennek kitalálása talán nem lesz nagy nehézség senkinek: **ASTAROT** ugyanis a nálunk lévő tárgyakkal együtt telepörtáltat bennünket a megadott helyszínre... A kalandjátékokban a farkasember kimondottan retteg az ezüsttárgyaktól – így van ez esetünkben is: ha a tárgy nálunk van, "Farkasfalva" (**WOLFDORP**) szobájában kóricálva nem kell tartanunk az utunkba akadó Werewolf-októl.

PELLET (galacsin): **GORBURG**-ban van (311) és a **NUGGET**-hez hasonlóan tudjuk megszerezni (ugyanolyan varázslat védí). A galacsint megvizsgálva, láthatjuk, hogy a "LJMAX" szó van ráírva. Ebből bizonyára mindenki könnyen rájön arra, hogy ez csakis Slug elleni védekezésre szolgálhat (az összefüggést mindenki találja ki maga). Slug ékes angol nyelven ugyan csigát jelent, de ez inkább egy maláchoz hasonlít, amely kabátot visel – őrá **SLYMOLE**-ban számíthatunk.

GARLIC (fokhagyma): **WOLFDORP**-ban találunk rá (133) és felvételekor megtudjuk, hogy ez nem étel. Erre nagyszerű megállapítás helyett inkább azt kellett volna elmondani, hogy mire szolgál – mindegy, később kiderült, hogy a **MORFANG**-ban lévő vámpírok ellen nyújt védelmet.

SLAT (léc): **MORFANG**-ban találhatjuk meg (127) egy ládában, amely előtt sajnos egy vámpír is álldogál (hagymát neki!). Az F jelű ajtó felé menet, **MIDUS**-ban (178), egy küklopsz állja az utunkat. Mielőtt megtámad bennünket, rögtön bele is szalad a lécebe és a továbbiakban egy szemmel kevesebbrel (azaz összesen nullával) néz halála elé.

SHELL (kagyló): **MIDUS**-ban gyűjthetjük be (165), és a mellékelt információ szerint az egy kagyló a meleg tengerből (**IT'S A SHELL FROM A WARM SEA**). A kagyló a róla elnevezett helyszínen (**ROOK OF HYDRA**) egyik szobájában (356) található hidra ellen szolgál. A kagyló talán nosztalgiaát ébreszt benne a régi szép "meleg tenger"-ben töltött idők után, ugyanis most egy cserépen üdögél a szerencsétlen (ergo ő nem mozog a szobákban). Ha a kagyló nálunk van, sértetlenül átgyalogolhatunk rajta (vigyázat, Axil a cserépnél megáll és elkezd elveszíteni életerőpontjait – tehát folyamatosan nyomnunk kell a megfelelő iránybillentyűt).

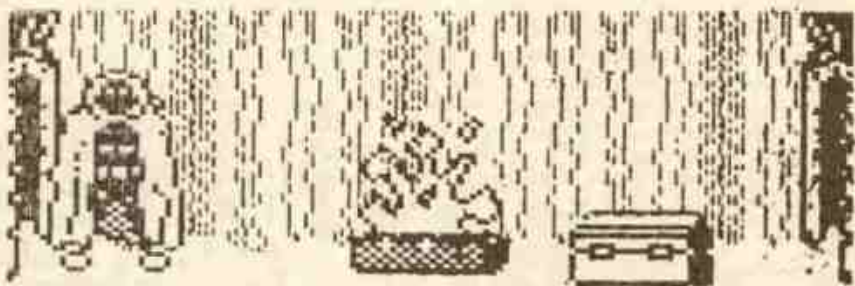


MIRROR (tükör): **TROOLWYND**-ban leljük rá (354). Az előző számban már említettük, hogy mire szolgál: a Medusa nevű hölgy támadása védhető ki vele, aki **ROOK OF HYDRA**-ban fog akadályozni bennünket (valószínűleg – a hidrához hasonlóan – ő is a meleg tengerből származik).

Vindicator • Imagine

A játék 2. szintjének kódja: VALSALVA MANOEUVRE

Ezidáig már bizonyosan okozott némi problémát, hogy a ROOK OF HYDRA 348 és a SOTHIC 265 helyszíneken némi akadály (nevezetesen a tűz és a víz) került utunkba – és nem is engedett tovább bennünket. A tűz esetében a megoldás kulcsa a CLASP (oszt). Egy ládában leihetünk rá TROLL-WYND-i bolyongásaink közepette (343). Miután felvettük, Axil megjegyzi, hogy "igazán eredeti dolog, olyan, mint egy szalamandra". A szalamandra úgy tűnik plexiből készülhetett, mert a tárgy birtokában nyugodtan átkelhetünk a tűzön (arra az időre elalszik). A víznél sajnos semmilyen tárgy nem segít rajtunk, tehát ismét a derék Apex-hez kell fordulnunk tanácsért (bár ennek eddig még mindig fejfájás lett a vége). Rendeljük tehát magunkhoz és érdeklődünk nála a vízről. Apex teljesen új oldaláról mutatkozik be, mert rögtön egy hasznos tanaccsal szolgál: "Beszélj le!". Hát akkor további szövegünket a vízhez intézzük: "WATER, DOWN" – hát ez nem jött össze. Akkor talán "WATER, FALL" – és lőn csoda, a szökőkútból vízesség lett. Ezt a ceremóniát egyébként a szobán áthaladva minden egyes esetben el kell játszaniunk.



A múltkor számban a D jeű ajtónál hagytuk abba a kulcsszavas ajtó ismertetését – most tehát folytassuk tovább az E-vel. Ez a 2. szint "Mennycsatorna Szeme" (EYE OF HEAVEN) helyszínéről nyílik, ahová három kulcsos ajtó kinyitása után tudunk elkeveredni. Ennek a legutolsó szobájában már régi ismerősként üdvözölhetjük az "örököt", akiktől rögtön megpróbálhatunk egy kis segítséget lejmölni ("GUARDS, DOOR"). Már jön is a válasz: "Szemet szemért, hogy bejuss a Paradicsomba" (AN EYE FOR AN EYE TO ENTER PARADISE). Puff nekil Hát kellett nekünk kérdezősködni...? Na mindegy, kezdetünk töprengeni – vagy begépelni az Ország-féle szótárban található összes szót. Az A-verzió pihentetőbbnek tűnik, inkább annál maradunk... Csapongó fantáziánk szálait – a Paradicsom kulcsszó alapján – ismét a vallás felé kell irányítanunk (már nem az első utalás rá a játék során). A Biblia szerint Ádámot és Ádámnéét (Éva néven is közismert) Isten elvtárs kilitította a "Paradicsom" nevet viselő aiberletűkből, mivel a kígyó felbujtására elfogyasztottak egyet a Tudás Fájának gyümölcséből (pedig az előzetesen megkötött bérleti szerződésben benne volt foglalva, hogy ebből csak a főbérlet fogyaszthat). Amikor az Isten tudomására jutott, kirúgta a fent nevezett két személyt a Paradicsomból ("Szemet szemért"), s csereelapként a lényegesen alacsonyabb nivót képviselő Földi Siralomvölgy nevű helyet utalta ki a számukra – valamint őket és utódaikat arra ítélte, hogy egész halandó létük során visszakiivánozzanak oda. Mint az isteni jóvendölések általában, ez is beteljesedett: itt állunk a HEAVY ON THE MAGICK közepén és minden vágyunk az, hogy bejussunk a Paradicsomba... Vágyódás, vágyódás... vágyódás? VÁÁÁÁGYODÁÁÁÁ!!! Ez az ajtó kulcsszava. Hogy maradjon valami a Kedves Olvasónak is az asszociációs mámorból, kikeresheti a szótárból az angol megfelelőjét...

A Paradicsomba belépve ugyanaz történik, mint a D jeű ajtó kinyitásakor: Axil egy érdekes népitáncot ad elő és nyolccal növekszik a tapasztalati pontjaink száma. Még érdekesebb a szöveg, ami közben a középső ablakban jelenik meg: "Jól van Axil, te nagyon Rátermett, ezt egy kijáratot csináltad!" (WELL DONE AXIL THE VERY ABLE YOU HAVE MADE IT TO AN EXIT). Aha – szóval éppen kijáratot gyártunk...

Az F jeű ajtó az 1. szinten található PILEFOOT egyik szobájából nyílik. Ez megint elég hosszú kórcátás árán közelíthető meg, valamint néhány kulcsot is használnunk kell előbb (nem is beszélve a kuklopszon végrehajtandó szemműtétről). Az örök által adott segítség most egész konkrétan tűnik – már annak, aki járatos a mágia területén: "A mágikus szavak száma" (NUMBER OF MAGICK WORDS). Mivel mi kevésbé voltunk tájékozottak, először úgy kezdtük, hogy megszámoltuk hány varázslatot tud Axil – nem jött be. Azután megszámloltuk, hány szót tud összesen – ez sem jött össze. Ekkor a kizárásos módszerhez folyamodtunk; feltételezve, hogy tort számú varázslat nincsen, elkezdünk szépen számolni az ajtónak ("DOOR, ONE; "DOOR, TWO... és így tovább). Ez olyan tíz tájékán már meglehetősen zsákutcának tűnt – de mi azért folytattuk. Aztán egyszer csak kinyílt az ajtó. Hogy hol? Hát azt mindenki kipróbálhatja maga is...

A G jeű ajtó a 3. szinten lévő TERTIA PORTA-ból nyílik, miután ájutottunk a tűzön és kinyitottuk az "Esők Szobáját" (ROOM OF RAINS). Az ajtó kulcsszaváról érdeklődve az Öröknél, valószínűleg megint a gutautés fog mindenkit kerülgetni: "Keress a tűzmadarat a bejutáshoz" (SEEK THE FIREBIRD TO ENTER DOOR). Hát akkor ti miért vagytok itt? Mindegy rendeljük magunkhoz Apex-et és érdeklődünk tőle a tűzmadár felől ("APEX, FIREBIRD"). "Ninos ilyen dolog" – érkezik a válasz ütődött barátunktól. Vagy éppen nem ez a neve? Apex-től a tűzmadár közismert mitológia; nevével tudunk információt kapni, miszerint meg kell keresnünk a fészket. Ide kell beleraknunk a WRAITHVALE-ben (252) található tojást (BONE), majd a fészkeknek megparancsolni, hogy legyen belőle tűzmadár. Ezzel azonban most nem foglalkozunk részletesen, mert az ajtó kinyitásához szükséges szót egy másik módon is megszerezhetjük, nevezetesen egy Ai nevű élőlénytől (pontosabban egyelőre még "halottlány"). Ez a módszer ugyan hosszabb, mint a tűzmadaras – viszont ismerete nélkül egy csomó tárgy feleslegesnek tűnik a játékban. (Egyébként jellemző a Gargoyle-mentalításra, hogy miután a tűzmadártól megtudtuk az ajtó kulcsszavát és elkezdünk Ai életre keltésével foglalkozni, úgy kb. fél órnyi szenvedés után tőle is ugyanazt a jelszót fogjuk megkapni...)

Ai-ról információt először "Ai Konyhájában" (KITCHEN OF AI) kapunk: a 328 szobában egy üstöt (CAULDRON) találunk. Elvinni ugyan nem tudjuk, de Apex-nál kérdezősködve róla, nagyszerű információt kapunk: "Egy fazék a kajának" (A POT FOR FOOD). Fantasztikus! Ki hitte volna? Mindegy, vizsgáljuk meg az üstöt: kiderül, hogy egy iratekeres taláható benne, amely az alábbi versikét tartalmazza (stílszerű fordítás mellékelve):

FOR AI IS DEAD
SEEK ARM, LEG, HEAD
IN POT DISPLAY
AND ONE WORD SAY.

"Mivel Ai barátunk teljes mértékben hulla,
Keress kezét, lábat, fejet (az egyik neve ULNA),
Pottyantsd őket üstbe, hogy mindet együtt lássad,
Már csak egy szót kell mondanod – kész a varázslat."

Ha már úgyis idehívtuk az előbb Apex-et, akkor rögtön meg is kérdezhettük tőle, hogy mi a véleménye Ai-ról ("APEX, AI"). "Hideg és halott" – feleli túlméretezett házbizondunk. Aha. Szóval Ai egy Málilita-csirkel! Kezdehetjük összeszedgetni a szükséges csontokat. A legnagyobb csontlelőhely a 4. szinten lévő METHOS. Itt meg is találjuk az általunk keresett dolgokat, név szerint:

- Sing csont (ULNA) – 465
- Combcsont (THIGH BONE) – 484
- Koponya (SKULL) – 461 (ez utóbbi úgy néz ki, mint a mérgezett fej).

Vindicator Imagine

A játék 3. szintjének kódja: EUTACHIAN TUBES.

Miután az **AJ**-puzzle 3 darabja nálunk van, ballagjunk vissza az üsthöz és dobáljuk bele őket (a biztonság kedvéért ezután vizsgáljuk meg, hogy minden darab benne van-e). Most kezdetünk próbálkozni, hogy milyen módon rángassuk vissza az élők közé a páciens: meg kell invokálni? (**INVOKE AJ** – nem nyert, ez csak a démonokra van hatással) hívni kell? (**CALL AJ** – ez sem jött össze, nem nagyon jött be az ajtón) netán egy szót kell mondanunk az üstnek (**"CAULDRON,AJ, CAULDRON, GENESIS, CAULDRON,FUCK OFF** és így tovább). Egy idő után rájöhettünk, hogy nem túl sok esélyünk van a kitalálására ezzel a módszerrel – nézzünk körbe inkább, hátha találunk némi segítséget valahol. Keresgélésünk tárgya lehetőleg a **GORBURG**-ban (315) megjelölhető könyvtár legyen – ezen ugyanis az áll: **ACHAD TO CAULDRON**, azaz **"ACHAD** az üstnek". Jé! Úgy látszik, a készítőknek ennél a résznél rossz napjuk volt – ez túl konkrét információ ahhoz, hogy igaz legyen. Pedig az – **"CAULDRON, ACHAD** és **AJ** measter visszatér közénk a Holtak Birodalmából. Vele is Apex-hez hasonló magasroptú beszélgetést folytathatunk, amit egy kissé unalmassá tesz az, hogy mindenre azt feleli: "csak az időmet pocsékolod" (**YOU WASTE MY TIME**). Egy kivétellel: **"AJ,DOOR"** hatására megkapjuk tőle a 3 ajtó kulcszavát.



Egyetlen ajtó maradt már csak hátra, a **H** jelű (majd később kiderül, hogy még van egy), ami a 4. szinten lévő Kriptából (426) nyílik. Ehhez a legnehezebb eljutni, mivel ide csak egyetlen út vezet – az, amelyiket a hidra állja el. Rajta átkecmerelve eljutunk a **PIT** helyszínelre, ahol némi botyongás után megtaláljuk az oroszlanos kulcs által nyitható Büszkeség Szobáját (**ROOM OF PRIDE**). Emögött a "Szakadék" (**CHASM**) nevű helyszín vár bennünket. Nemsokára kiderül az is, hogy honnan kapta a nevét: a 446 szobában ugyanis egy szakadék állja az utunkat, amelyre belelöpve **Axil** megkezdte az "Utazás a Föld középpontja felé" című játékot (ez azonban egy teljesen más játék – jelen programban ez a **GAME OVER** nevű jelenségét vonja maga után). A falon egy érdekes jel is látható, amelyel eddig nem találkoztunk. Magunkhoz rendelhetjük Apex-et némi segítség reményében (a társalgás jegyzőkönyvétől ezúttal eltekintünk), de nem sokra fogunk jutni vele. A megoldás kulcsa itt ugyanis nem a logikában (már amennyiben valaki találta ilyen ebben a játékban...), hanem a szorgalomban van: a leírás elején említettük, hogy minden tárgyról információkat szerezhetünk azzal, hogy megvizsgáljuk, felelőssük vagy Apex-et kérdezzük róla. Ha ezt minden tárggyal eljátszottuk, akkor már rábukkanhattunk a megoldás kulcsára: **MORFANG** egyik szobájában (118) ugyanis találhatunk egy üveget (**FLASK**), amelyet felvéve azt az információt kapjuk, hogy egy római négyes és egy húszas szám található rajta (**IT'S INSCRIBED WITH THE WORD IV AND TWENTY**). Márpedig a 420 – a program által használt módszer alapján – pontosan megegyezik annak a szobának a koordinátáival, ahol a szakadék állja utunkat. Ha az üveg nálunk van, akkor baj nélkül átkelhetünk rajta (valamint plusz szerencsepontot is eredményez).



Megint egy kis séta következik: a "Kukacgyűrű" (**WORMRING**) körbesétálva (nem árt, ha viszünk magunkkal egy pénzeszsákokat is) és néhány utunkba akadó **Wyvern** leküzdése után (a 451 szobát kerüljük ki, mert az ajtón belépve pontosan rálépünk az egyikre) kinyithatjuk a **ROOM OF SCALE** szobát. Már is a "Rabak kétsége" (**DOUBT OF RABAK**) nevű helyen találjuk magunkat. Továbbhaladva az egyik helyszínen egy szörnyű rondságba (valami viziló-féleségbe) ütközünk. Valószínűleg ő lesz az a bizonyos **Rabak**, akinek kétségei vannak. Mondjunk neki valamit. Kiderül, hogy "Találós kérdés"-t akar velünk játszani: "Mondd meg a keresztnevedet" (**SAY MY LAST NAME**). Aha, szóval ezért vannak kétségei – elfelejtette a nevét. Talán kérdezzük meg tőle, hátha eszébe jutott közben (**"RABAK, NAME"**) "Egy sárkány vagyok" (**I AM A DRAGON**) – mondja. Játsszunk vele visszhangot (**"RABAK,DRAGON"**) "Melegsél" (**YOU ARE WARM**) – jön a válasz. **"RABAK,WARM** – "Mondd meg az utónevem". Szóval visszajutottunk ugyanoda. Próbálkozzunk egy másfajta megközelítéssel, mondjuk neki, hogy tűz (**"RABAK, FIRE"**) "Kétféle azt" (**I DOUBT IT**) – mondja. Azt tudjuk, arról vagy híres. Esetleg ennek az ellentéte? **"RABAK,WATER** – volt **Rabak**, nincs **Rabak**. Mehetünk tovább. A következő helyszínen kellemetlen meglepetés fogad bennünket: három oszlopot láthatunk, rajtuk a kulcszavas ajtónál megszokott örökkel – de nincs közöttük ajtó! Hát hol van? **"GUARDS,DOOR** és már kapjuk is a következő elmés versikét:

**NO SOUTH DOOR, WHAT A SHAME,
SEEK ERLSTONE FOR THE NAME.**

"Ó, micsoda szégyen, nincsen ajtó délre!
Hát keresd az Erlstone-t – és rájössz a névre."

Tessékék? Mi az az Erlstone? Az odáig rendben van, hogy annyit jelent "Tündérek Köve" – ilyen tárggyal azonban nem találkoztunk a játék egyetlen szobájában sem! Nem tud róla sem Apex, sem a tárgyügyekben illetékes két démon sem! Akkor hol az ördögben keressük? Mert nekünk mindenképpen át kell itt jutnunk, hiszen emögött van a **H** jelű ajtó! Mivel semmi más támpontot nem találtunk és fogalmunk sem volt, hogy az Erlstone-t merre keressük, kénytelenek voltunk a hely nevének a szinonimáival próbálkozni. Reménytelen próbálkozás volt – mégis sikerre vezetett! Az ajtó kulcszava: "Halottak Kapuja" (a "halott" szó ráadásul egy olyan kifejezés volt, amit a jelen angol nyelv már nem is használ – elavult formula!). Úgyan arról fogalmunk sincsen, hogy mi köze van Erlstone-nak a Halottak Kapujához, mindenesetre az ajtó kinyílt rá. Na, hogy hangzik a kulcszó angolul...?

Innen már az Olvasóra vár a továbbjutás. Annyit mindenesetre elárulhatunk, hogy a **H** jelű ajtó (az örök rejtvénye és annak a megfejtése roppant elmés...) mögött is ugyanaz a jelenség játszódik le, mint a **D** és **E** ajtókon való belépéskor (népítánc, villogás, "Jól van, **Axil**...", stb.). Itt azonban már mi is elakadtunk: minden tárgyat felhasználítottunk illetve tudjuk, mire szolgálnak (a nugát (**NOUGAT**), ami nem étel és a feleslegesnek ítélt tárgyak kivételével), minden zárt ajtó mögé bejutottunk és bejártunk a 256 helyszínből 255-öt. Egy helyre nem sikerült bejutnunk: a 488 szobába ugyanis – a **FURNACE**-hez hasonlóan – sehonnan sem vezet ajtó. Így tehát nyitva maradt néhány kérdés, amelyre nem tudunk választ adni, nevezetesen:

- mi van a 488 szobában és hogyan lehet oda bejutni?
- micsoda vagy micsoda az előbb említett Erlstone és hol található?
- mi a játék vége? Az eddigiekből ugyan kiderült, hogy kijáratot kell gyártanunk, de nem tudjuk, hogy hol lesz ez a kijárat (talán a 488-ban?) és mit kell még csinálni hozzá? Ha valaki ismerné a választ ezekre a kérdésekre, kérjük küldje el az SpV címére, hogy közöthessük (egyébként minket is furdal a kíváncsiság)! Hát akkor annyit a **Gargoyle**-játékokról...



SPECTRUM programok átírása 6.

Ha valaki vette a fáradságot és előállította a program két blokkját, valamint átalakította a közötti betöltőt, csodálkozva tapasztalta, hogy a program nem igazán rendeltetésszerűen működik. Általánosságban elmondhatjuk, hogy ha egy program "élszáll", akkor annak két oka van:

1. A SPECTRUM ROM-ban lévő szubrutint hív
2. Megszakítást alkalmaz.

Esetünkben mindkét esetre látunk majd példát, most foglalkozunk a fontosabbal, a felhasználói megszakítással. A megszakítás olyan alprogram, amely a főprogram futása közben egy előre meghatározott esemény bekövetkeztéig hajtódik végre. Ez az esemény esetünkben esetlegesen esik meg (jó mi?) mégpedig minden 1/50-ed másodpercben. A megszakítás elfogadásakor a processzor elmenti az aktuális programszámlálót (a PC regisztert), majd reagál, az üzemmódtól függően más és más módon.

A Z80-as mikroprocesszornak 3féle megszakításkezelése van:

1. A nullás üzemmód. Ekkor a processzor a következő utasítás-byte-ot nem az aktuális memóriacímről veszi, hanem a megszakítást generáló eszköz adja. A Z80 felhasználók ezt a módszert szinte nem is használják, az INTEL 8080 mikroprocesszorral való kompatibilitás miatt építették be. Beállítás az "IM 0" Z80 utasítással történik.
2. Az egyes üzemmód. Ekkor végrehajtódik egy "RST 38H" utasítás, vagyis meghívja a 38H-n lévő szubrutint. A SPECTRUM BASIC ezt használja. (Valamint az ENTERPRISE EXOS is!) Beállítása egy "IM 1" (ED 5E) utasítással történik.
3. A kettes üzemmód. Ez a legcifrább, de a leghasznosabb is. A megszakítás alprogram címének megtalálásához meg kell néznünk egy speciális Z80 regisztert, az "Interrupt vector register" névre hallgató "I" regisztert. Ebben az üzemmódban ugyanis megszakítások esetén a mikroprocesszor beolvas az adatvezetésekről egy byte-ot, a felső 8 bitnek megfelelően az "I" regiszter 8 bitjét, majd az így létrehozott 16 bites címről kiolvass egy byte-ot, majd a 16 bites címet megnövelve még egyet. Az első byte lesz az alprogram címének első, a második pedig a felső byte-ja. Mire jó ez? Például arra, hogy ha több egység kérhet megszakítást, akkor nem a processzor kezdi el találgatni a megszakítani kívánó személyét, hanem a periféria azonosítja magát az adatbuszra küldött byte-al. A beállítása az "IM 2" (ED 5E) utasítással történik. Ezt az utasítást mindenki jól jegyezze meg, ugyanis a programok 80-90 százaléka ilyen üzemmódot állít be.

FONTOS! A megszakítást kiszolgáló alprogramnak nem szabad megváltoztatni EGYETLEN regiszter értékét sem, amelyet a főprogram használ! A megszakítás elfogadásakor tiltódik a további megszakítások elfogadása, ezt általában az interrupt alprogram végén szokták engedélyezni. Ez mind nagyon szép – mondhatja az Olvasó – de mi szükség van ennek ismeretére? Nagyon is van, mivel az ENTERPRISE bonyolultabb megszakítás elektronikával rendelkezik. Nem egy-, hanem mindjárt négyféle megszakítás forrás áll rendelkezésre, amelyek külön-külön tiltathatók, engedélyezhetőek. Erre egy külön perifériacím szolgál, melynek címe a 0B4H. Ez különbözik OUT és IN műveletek esetén. Minket csak az OUT érdekel, ezért csak ezt részletezzük.

A port bitenként értelmezzük:

- b0 A hanggenerátor által adott megszakítás engedélyezése. (1 érték által)
- b1 A hanggenerátor megszakítás tároló törlése.
- b2 Az 1 Hz-es megszakítás engedélyezése.
- b3 Az 1 Hz-es megszakítás tároló törlése.
- b4 A video-megszakítás engedélyezése.
- b5 A video-megszakítás tároló törlése.
- b6 A soros vonal megszakításának engedélyezése.
- b7 A soros vonal megszakítás tárolójának törlése.

Az, hogy az engedélyezésnek mi a szerepe, elég egyértelmű: ha a megfelelő helyre egy-egy egyes bitet írunk, akkor az a forrás képes

megszakítást generálni. Ha nullát írunk, akkor letiltottuk az eszközt. Az 50 Hz-es megszakítás generálásához célszerű a NICK-chip által generált megszakítást használni, amit az LPT generálásakor beállítottunk.

A másik feladat – a megszakítás tároló törlése – már nem ilyen egyértelmű. A DAVE-chip (amely a megszakításrendszert táplálja) tárolja az interrupt kéréseket. Amíg nem töröltük a tárolóját, addig elnyom minden ilyen irányú próbálkozást. Tehát: ha azt akarjuk, hogy az 1/50 másodpercenként érkező megszakítások ne vesszenek el (vagyis ne "fagyjon le" a program), minden megszakítást kiszolgáló alprogram végén (vagy elején) törölünk kell a DAVE-chip megfelelő tárolóját. A video-megszakítás újraengedélyezését végző rutin az alábbi:

```
LD A,30H ;b4 + b5 video IT. engedélyezés, tároló törlése  
OUT (0B4H),A ;A DAVE-chip megfelelő regisztere
```

Arra természetesen figyeljünk, hogy mentsük az "A" regiszter a művelet előtt.

Most nézzük meg, hogy a nagy fáradsággal átmentett "MOON CRESTA" hogyan csinálja. A program indítási címe 77B2H.

77B2	F3	D1				:Megszakítás tiltása.
77B3	31 43	F6	LD	SP,F643		:Verem állítás.
77B6	21 10	A7	LD	HL,A710		:Karakterkészlet helyének tárolása az 5C36H címen.
77B9	22 36	5C	LD	(5C36),HL		:A betöltőkép tárolása
77BC	21 05	40	LD	HL,4005		
77BF	11 7B	5C	LD	DE,5C7B		
77C2	01 FB	1A	LD	BC,1AFB		
77C5	ED 80		LDIR			
77C7	21 00	F5	LD	HL,F500		:Interrupt cím beállítása.
77CA	11 01	F5	LD	DE,F501		
77CD	01 01	01	LD	BC,0101		
77D0	36 77		LD	(HL),77		:A megszakítási rutin 7777H-n.
77D2	ED 80		LDIR			
77D4	3E F5		LD	A,F5		:A táblázat felső byte-ja (F5H) az "I" regiszterben.
77D6	ED 47		LD	I,A		:Kettes mód beállítás.
77D8	ED 5E		IM	2		:Megszakítás engedélyezése.
77DA	FB		EI			

Biztos mindenki megdöbben a 257 elemű táblázaton. Erre azért van szükség, mivel a táblázat alsó byte-ja (amit a periféria ad) nem elégséges biztos. (Általában FFH) Ezzel a módszerrel minden kombinációra bebiztosítottuk magukat a programozók. Az ilyen táblázatos módszerek leggonoszabbja az, amikor a táblázat csupe 255-ből áll, tehát a megszakítási rutin címe 85535 (FFFFH). Erre az egy byte-ra leraknak egy "JP" utasításnak megfelelő 18H-t, amelynek operandusa a 0 címen lévő (SPECTRUM BASIC) 0F3H. Ez a relatív ugrás visszaugrik a 0FFF4H címre, ahol egy ugró utasítás visz a tulajdonképpeni megszakítás alprogramra. A következő LOADER a kész játék végső betöltője, ezért érdemes beírni. A megszakítási problémát úgy küzdöttük le, hogy 7777H-ra (az IT rutin) letettünk egy "JP" utasítást, előbb engedélyezzük a DAVE-chipben az interruptot, majd ráugrunk a tulajdonképpeni rutinra.

```

ORG 256
JP CONT
JP KEY
JP ALLKEY
JP JOY
JP BEEP
KEY EX (SP),HL
LD A,(HL)
INC HL
EX (SP),HL
OUT (0B5H),A
IN A,(0B5H)
RET
ALLKEY PUSH BC
LD BC,0AFFH
A1 LD A,B
DEC A
OUT (0B5H),A
IN A,(0B5H)
AND C
LD C,A
DJNZ A1
POP BC
RET
JOY PUSH BC
LD B,0
CALL JOY1
LD A,C
LD (JOY2+1),A
CALL JOY1
LD A,C
JOY2 OR 0
POP BC
RET
JOY1 LD C,B
JOY3 LD A,B
INC B
OUT (0B5H),A
IN A,(0B6H)
RRA
CCF
RL C
JR NC,JOY3
RET
BEEP LD A,L
SRL L
SRL L
CPL
AND 3
LD C,A
LD B,0
LD IX,L03D1
ADD IX,BC
LD A,(L5C48)
AND 3BH
RRC A
RRC A
RRC A
OR 0
L03D1 NOP
NOP
NOP
INC B
INC C
L03D6 DEC C
JR NZ,L03D6
LD C,63
DEC B
JP NZ,L03D6
XOR 1BH
OUT (0A7H),A
LD B,H
LD C,A
BIT 4,A
JR NZ,L03F2
LD A,D
OR E
JR Z,L03F6
LD A,C
LD C,L
DEC DE
JP (IX)
L03F2 LD C,L
INC C
JP (IX)
L03F6 EI
RET
L5C48 EQU 5C48H
CONT LD SP,STACK
LD B,1
LD C,2B

```

L1

L2

```

EXOS 6
LD A,1
EXOS 3
LD A,48
OUT (0B4H),A
LD A,195
LD HL,ENABLE
LD (7777H),A
LD (7778H),HL
LD HL,ISR
LD (38H),A
LD (39H),HL
LD HL,JOY
LD (28H),A
LD (29H),HL
LD A,20H
OUT (0A8H),A
OUT (0ACH),A
XOR A
OUT (0B1H),A
LD HL,4000H
LD DE,STACK
LD BC,4000H-STACK
LDIR
LD A,0FDH
OUT (0B1H),A
JP 77B2H
ISR PUSH AF
LD A,30H
OUT (0B4H),A
POP AF
EI
ENABLE PUSH AF
LD A,30H
OUT (0B4H),A
POP AF
SCREEN JP 0EE00H
ALSO DEFB ALSO-$-1
DEFM *CRESTA.SCR*
FELSO DEFB FELSO-$-1
DEFM *ALSO*
LINE DEFB LINE-$-1
DEFM *FELSO*
SYNC DEFB 255,14H,15,2FH,0,0,0,0
DEFB 0,32,72,104,144,176,216,248
DEFB 0,0,0,0,0,0,0,0
DEFB 247,8,11,73H,0B8H,0FEH,0E9H,1
DEFB 0,216,216,0,0,0,0,0
DEFB 217,12H,63,0,0,0,0,0
DEFB 0,0,0,0,0,0,0,0
DEFB 253,16,63,0,0,0,0,0
DEFB 0,0,0,0,0,0,0,0
DEFB 252,16,6,63,0,0,0,0
DEFB 0,0,0,0,0,0,0,0
DEFB 255,90H,63,32,0,0,0,0
DEFB 0,0,0,0,0,0,0,0
DEFB 252,12H,6,63,0,0,0,0
DEFB 0,0,0,0,0,0,0,0
DEFB 207,13H,63,0,0,0,0,0
DEFB 0,0,0,0,0,0,0,0
HOSSZ EQU $-SYNC
STACK EQU $+200

```

Mint látható, a program két blokkját nemcsak egyszerűséggel csak "ALSO" és "FELSO" névvel illettük, ezáltal is jelezve az elhelyezkedésüket. A betöltő elején található rutinok méltatására a későbbiekben visszatérünk. Ezzel a betöltővel futtatva a "MOONCRESTA"-t már hajlandó elindulni. (Igaz sok köszönet nincs benne!)

És most essék néhány szó a SPECTRUM billentyűzetfigyeléséről: Az összes gomb egy mátrixba van kötve. A mátrix sorát mi közöljük a géppel igen ravasz módon, az oszlopot pedig szerény keresünkre válaszként kapjuk. A sor kiválasztását igen érdekesen oldották meg. Kihasználták a mikroprocesszor ama tulajdonságát, hogy IN/OUT műveletek esetén a címbusz felső 8 bitjén is értékes adat van. Vegyünk példának egy IN A,(0FEH) utasítást!

Ekkor a címbusz alsó 8 bitje a periféria címét (jelen esetben a 0FEH-t), míg a felső 8 bit az "A" regiszter beolvasás előtti értékét tartalmazza. A másik fajta "IN" utasításcsoport esetén (az IN r,(C)) a "B" regiszter kerül hasonló helyzetbe.

A SPECTRUM a billentyűket egy 8*5 elemű mátrixban kezeli. A 8 sor eléggé sokatmondó, mivel pontosan ennyi bit van egy byte-ban. A 8 sornak tehát megfelel 8 bit, amelyik bit 0 értékű, azt a sort olvassa be a processzor. Mivel egy byte-on belül bármelyik bit lehet 0, ezért egy utasítással több sort is lehet kérdezni. A kiválasztott sort a 254-es (0FEH) porton lehet beolvasni, az alsó 5 bit hordozza a billentyűzet információit. Konkrétan: az

1,2,3,4,5 gombok vizsgálata az alábbi kis programocskával történhet.

1.variáció	2.variáció
LD A,0F7H	LD BC,0F7FEH
IN A,(0FEH)	IN A,(C)

Működés szempontjából a két módozat egyenértékű, a 2. variáció azonban használja a "BC" regiszterpárt. A gyakorlatban mindkét variáció előfordul.

Ha billentyűfigyelést keresünk egy programban, akkor a következő utasításokra figyeljünk fokozottan:

Utasítás	HEXA kód
IN A,(0FEH)	DB FE
IN A,(C)	ED 78
IN B,(C)	ED 40
IN C,(C)	ED 48
IN D,(C)	ED 50
IN E,(C)	ED 58
IN H,(C)	ED 60
IN L,(C)	ED 68

Mielőtt bárki azt hinné, hogy az első két utasításon kívül más nem használnák, az nem jár messze az igazságtól, viszont elretentésül megjegyeznénk, hogy leltünk már IN C,(C) utasítást is, amely az irányítás nemes feladatát volt hivatva betölteni. Gyakran előfordul a teljes kiaviatúra lekérdezése is. Ez SPECTRUM-on egyszerű, ENTERPRISE-on viszont már nem. (Ha valaki nem tudná, egy

```
XOR A
IN A,(0FEH)
```

nagyon megfelelő ilyen célokra)

A téma lezárásaként itt van a sokat emlegetett mátrix.

A SPECTRUM billentyűzetmátrix:								
BIN	DEC	HEX	1	2	4	8	16	
			b0	b1	b2	b3	b4	
11111110	254	FE	Caps	Z	X	C	V	
11111101	253	FD	A	S	D	F	G	
11111011	251	FB	Q	W	E	R	T	
11110111	247	F7	1	2	3	4	5	
11101111	239	EF	0	9	8	7	6	
11011111	223	DF	P	0	I	U	Y	
10111111	191	BF	Enter	L	K	J	H	
01111111	127	7F	Space	Sym.	M	N	B	

Miután a SPECTRUM kiaviatúra-kérdést ilyen jól helybenhagytuk, rátérhetünk az ENTERPRISE oldalára. A mátrix módszer itt is alkalmazott, viszont — a több gomb miatt — némileg eltér a fent említettől. Először is, nem 8 sor van, hanem 10. Ez előrevetíti ama szomorú tény árnyékát, hogy nem fér egy byte-ban (vagyis csak a külön bites módszerrel nem). Másodsor: nem 5 oszlopot olvasunk be, hanem 8-at. Ezzel a módszerrel 80 gomb figyelmére van mód, ebből néhány nem használt. Azt, hogy melyik sort (0-tól 9-ig számozva) akarjuk olvasni, a 0B5H című porton kell közölnünk. Miután ezt közöltük, rendelkezésünkre áll ugyanezen a porton az adat. Előző példánkban maradványok az 1-5 billentyűket az

```
LD A,3
OUT (0B5H),A
IN A,(0B5H)
```

utasításokkal figyelhetjük. Látható, hogy az előbbi 4 byte-tal szemben most ugyanez 6 byte hosszan sikerült. Vagyis nem fér be az eredeti helyére. Itt három dolgot tehetünk:

1. Hagyjuk az egészet és elmegyünk dominózni.
2. A memória egy nem használt helyére megírjuk szubrutinként, majd az eredeti programrészt lecseréljük egy "CALL" utasításra.
3. Addig erőltetjük (jó VOGON szokás szerint) amíg nem sikerül begyűrni az eredeti helyére.

Kétségkívül az első módszer a legegyszerűbb, viszont a harmadik a legszimpatikusabb, ennek ellenére mi a másodikat fogjuk bemutatni. Ennek okai: minek riasszuk el az Olvasót ilyen korai stádiumban?

Végezetül íme, az ENTERPRISE billentyűzetmátrix:

Sor	b7	b6	b5	b4	b3	b2	b1	b0
	12B	64	32	16	8	4	2	1
0	80H	40H	20H	10H	08H	04H	02H	01H
0	B.SH.	Z	X	C	V	B	\	N
1	CTRL	A	S	F	D	G	LOCK	H
2	TAB	W	E	T	R	Y	Q	U
3	ESC	2	3	5	4	6	1	7
4	F1	F2	F7	F5	F6	F3	F8	F4
5		ERASE	-	0	-	9		8
6			:	L	:	K		J
7	ALT	ENTER	BAL	HOLD	FEL	JOBB	LE	STOP
8	INS	SPACE	J.SH.	.	/	,	DEL	M
9			P	@	0			1

GUNRUNNER

A messzi jövőben játszódó programban hősünknek komoly feladat jutott! Egy szál pisztollyal végig kell küzdenie magát mindenféle veszéllyel tele-tűzdelt pályák sokaságán. Mindez csakis egy cél érdekében történik: el kell pusztítania a rejtélyes, oszlopokra állított fekete gömböket! Ebben különféle cselesen repülő műty-műtyök akadályozzák leginkább. Szerencsére segítségére vannak különböző megszerezhető tárgyak:

Jetpac: Ezzel a hátra csatolható repülő szerkezettel gyorsan átrepülhetünk az akadályok felett.

Shield: Pajzs, mely a repülő tárgyakkal való ütközés hatásaitól óv meg bennünket.

Poison: Egyfajta gránát, amely elpusztítja a repülő tárgyakat, valamint a felénk dobált kis gránátokat.

Multi fire: Megháromszorozza fegyverünk tűzerjét.

Sajnos ezen tárgyakkal van egy szépséghibájuk is, csak egy ideig hatásosak! Hogy meddig, azt a pontszám mellett lévő nevük sárga elszíneződése mutatja. Minél sötétebb a felirat, annál inkább lehet számítani a hatásosságuk megszűnésére. Csak a **multi fire** "örökéletű", de ezt is elveszthetjük — a többi tárggyal egyetemben —, ha az idegen repkedő tárgyakkal sokáig érintkezünk!

Egy-egy szint teljesítése után egy **BONUS** pályát teljesíthetünk időre, ahol halálunk esetén nincs életvesztés, csak jutalmul kevesebb pont úti a markunkat.

A játékot a már ismert módon a menüben beállított módon mind külső, mind belső botkormánnyal, valamint billentyűzetről is játszhatjuk. Előbb állítsuk be a billentyűzetet, mert csak így tudjuk a **poison**-t aktivizálni. Igen élvezetes piff-puff játékkal lettek gazdagabbak az ENTERPRISE tulajdonosok!



SCORE
4750

JET
POISON

PAC
SHIELD

LIVES
0

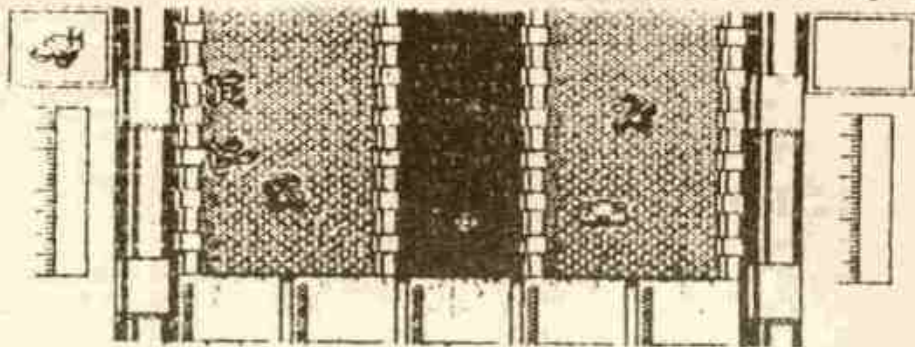


UCM (THE ULTIMATE COMBAT MISSION)

Ez az egyik legújabb SPECTRUM/ENTERPRISE átírat. Végre egy olyan program, amelyet nem csak szülőben lehet játszani!

A betöltést követően a menüből kiválaszthatjuk a játékosok számát, és az irányítást is. Ezt követően már csak a rossz bácsik elleni botkormány-gyilkos küzdelem van hátra.

A pályát felülről látjuk, a képernyő két oldalán az általunk irányított emberkéjünk energiáját láthatjuk. A képernyő alján a megülvő életelink számát jelképező figurák láthatóak. Sajnálatos módon játékosonként csak 3 élettel rendelkezünk. A játék célja, hogy elérjük a pálya végét és az eredményes küldetés jutalmául feliratkozhatunk a ponttáblázatba. Ez nem is olyan egyszerűen végrehajtható feladat örökélet nélkül. Az emberkének végtelen lőszer áll rendelkezésre a géppisztolyhoz, és időnként a NINJA kinézetű rossz alakoktól is begyűjthetünk pár rakétavetőt. A dolgok szépséghibája az, hogy amikor elhalálozunk, a begyűjtött fegyverünk is elvesz, marad a géppisztoly. A játék végigjátszása sok ügyességet és türelmet kíván. Vigasztaljon bennünket az a tudat, hogy a botkormányt javító szervezetek felkészülten várják a játék intenzív művelőit.



UP & DOWN



A szélesszájú kisbéka egy napon a mezőn ugrándozott, és Rádiós Walkman-jén a Calypso rádiót hallgatta. Olyannyira belefeledkezett az élet dolgiba, hogy egyszer csak egy nagy üregbe esett. Meglepetésében csak annyit tudott mondani, hogy: "Játék! Hahahahaaaaaa...", és máris a barlang fenekén találta magát. Itt a nagy zuhanás okozta utódéstől kiestek az elemek a Walkman-ből, és a szélesszájú kisbéka máris a rádióelvonás tünetei kezdtek jelentkezni, érthető hát, minél előbb a felszínre szeretett volna keveredni. Sajnos szegényke nem emlékezett a felfelé vezető útra – nagyon, nagyon buta szélesszájú béka volt –, ezért segítségre van szüksége! A bal- és jobboldali "SHIFT" billentyűkkel tudjuk brekkenszűnket irányítani. Minél hosszabb az ugrás erősségét jelző sáv, annál nagyobbat ugrik barátunk.

A breki nem csak butácska, hanem ügyetlenke is, ezért a gombokon és egyéb gizgazokon könnyen megcsúszik, ennek várhatóak a következményei: visszapotyan a mélybe.

Sajnos az üreg lakója, akiről nem lehet pontosan tudni, hogy kicsoda-micsoda lény – talán leginkább egy madárba oltott sárkányra hasonlít – lelkesedik a franciás kosztért, a Walkman-es szélesszájú kisbékaért. Érthető hát, hogy mindent megtesz annak érdekében, hogy kis zöld barátunkat megkaparintsa! Erre két oka is van: egyrészt ő is szeret rádiót hallgatni, másrészt igen éhes az istenadta! A lényeg az, hogy jó messzire elkerüljük, és a felszínre jutva elkálthassuk magunkat, miszerint: "Rádióóó Calypsóóó! Brekkk, na ugye?"

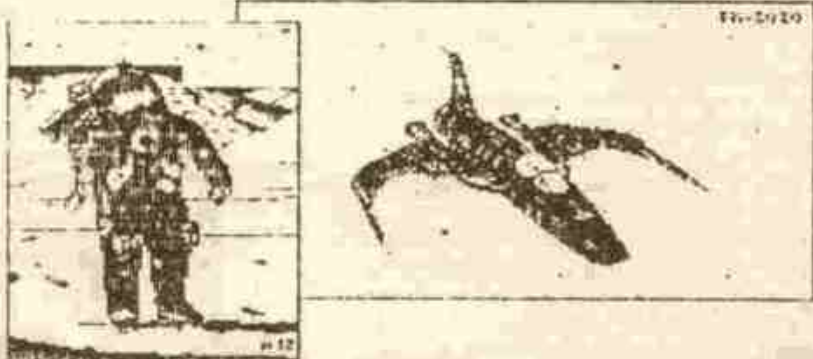
THUNDERCEPTOR

Azt hittuk, hogy az űr-földöldözős játékokban már semmi újat nem tudnak kitalálni, hiszen régen lerágott csont már ez a téma. Mégis kellemesen csalódtunk, mert ez a játék grafikailag kiváló és a képernyő kezelését tekintve meglepő gyorsasággal dolgozik. Minden elismerésünk a programozó(-k)nak!

A játék betöltődése után a menüből választhatjuk ki az irányítást, de vigyázzunk, mert indítás után ez a menüpont már többet nem tér vissza. Az irányítás kiválasztása után a játék következő menüpontjai ikon-rendszerből választhatók ki, egy mozgatható nyíl és a tüzelés segítségével.

Lehetőségünk van a fegyverzethez tartozó löszerek, illetve a védőpajzs erejének a megváltoztatására. Megnézhetjük a kitüntetésünket, illetve egy-egy pálya teljesítése után lementhetjük a játékot. Megnézhetjük, hogy mennyi és milyen típusú ellenséges űrhajót lőttünk le.

A játékban csak az ellenség löhető le, a szembe száguldozó meteoritok nem! Ez utóbbiak sokszor okoznak problémát, mivel nem onnan bukkannak fel, ahonnan várjuk őket. Autofire-s botkormány nélkül meglehetősen nehéz a normál löszerral való harc. Tudniillik a gombismétlés tiltva van, így azután nyomorekka nyomhatjuk a tűzgombot. Ebben a játékban még nem sikerült az utolsó pályáig eljutni (még örökélettel sem), ha igaz a küldetések számát jelző szám, akkor 999 999 pálya van. A játék végigjátszásához nem örökéletet, hanem végtelen türelmet kívánunk!



micro-PROLOG

Gépi kódú bővítés

A tárfelosztás

A micro-PROLOG az alábbi vázlat szerint helyezkedik el a tárban, illetve használja azt:

0	16384	24576	38767	38912	40357	65534
ROM	PROLOG	nem használt	munka-terület	kupac	verem	
0h	4000h	6000h	976Fh	9800h	9DA5h	FFFEh

A program a 9760h-n kezdődő néhány utasításra csak induláskor adja a vezérlést – három rendszerváltóztól (MODE, ATTR_P, ATTR_T) alaphelyzetbe állít –, majd a 6003h címre ugrik. A 976Fh-97Ffh területet nem használja; erre a 145 byte-ra pl. kényelmesen betölthető grafikus karakterként (UDG) az összes hiányzó magyar ékezetes betű. A 9800h-9DA4h címtartomány munkaterület, a 9DA5h-FFFEh dinamikusan kezelt kupac, illetve verem. A kupac (heap) alulról felfelé bővül, a verem pedig felülről lefelé. **Hulladékgyűjtő eljárás** (garbage collection) szabaddítja fel időnként a már nem használt területet (a kupac miatt erre szükség van). Ezt a felhasználó is kérheti a SPACE relációval, de általában a rendszer automatikusan elvégzi, ha fogytán a felhasználható memória.

Gépi kódú relációk számára a kupac kezelőcímének áthelyezésével foglaltatunk helyet. Ez a cím 9666h-n található (9DA5h), ezt az értéket kell a szükséges hosszúsággal megnövelni. Ha tehát n byte-ot akarunk lefoglalni, akkor 9666h-ra 9DA5h+n írható, s ezzel rendelkezésünkre áll a 9DA5h-(9DA5h+n-1) címtartomány. Ha e területre olyasmit írunk, amit az interpreter is kezel, akkor az esetleges törítés ellen a 7857h-n lévő, eredetileg szintén 9DA5h értéket ugyanúgy kell módosítanunk, mint a 9666h-n lévő (erre nincs szükség, ha pl. UDG-t teszünk oda).

Kisebbségi rutinok elhelyezhetők még a már említett 976Fh-97Ffh nem használt területen, illetve a rendszer 6000h kezdőcíme előtt, a betöltő BASIC program után.

Mint már láttuk, a micro-PROLOG koncepciójának egyik alapja a bővíthetőség. A micro-PROLOG-ban elkészített programok olyan relációk, amelyek a rendszert bővítik, beleértve ebbe a rendszer szintaktikáját is (pl. SIMPLE). Előfordulhat azonban, hogy nem a nyelv szelleméhez illeszkedő programra lenne szükségünk – pl. bonyolult aritmetikai számítások, perifériakezelés stb. –, ilyenkor a PROLOG-ban megírt relációk nehézségek, hely- és időigényesek. Szerencsére a nyelv, viszonylag egyszerű módon, gépi kódú relációkkal is bővíthető, amelyek a rendszerbe illesztésük után a rendszer relációkkal azonos módon használhatók. A micro-PROLOG és a hozzá illesztendő gépi kódú program közötti kapcsolat három részből áll. Az első ezek közül az ún. adatregisztereken keresztül megoldható paraméter átadás. A második a paraméterfa definiálása, amely a paraméterek lehetséges típusait, illetve típuskombinációt, s a gépi kódú program ezeknek megfelelő kilépési pontjait írja ki. A harmadik a gépi kódú programok, mint micro-PROLOG relációknak az elnevezése és a rendszer-relációk listájába illesztése.

Szintaktikus egységek

A micro-PROLOG-gal való kompatibilitás érdekében bevezetünk néhány ott használt fogalmat és ismertetjük fizikai ábrázolásukat.

Értéksajt

Három byte-os egység, amelynek első byte-ja egy típust, rákövetkező két byte-ja pedig egy értéket, vagy egy értékének a címét tartalmazza.

Az értéksajtok a rendszer különböző elemeinek az építőkövei; segítségükkel adhatók meg a számok, szövegek konstansok és a listák. A legfontosabb típusok a következők:

Típus	Az értéksajt első byte-ja	Az értéksajt második-harmadik byte-ja
Binárisan tárolt szám	4 (fh)	Maga a szám, kettes komplementum formában (Az aritmetika olyan, hogy csak -32767-től 32767-ig terjedő értékeket vehet fel)

Típus	Az értéksajt első byte-ja	Az értéksajt második-harmadik byte-ja
Lebegőpontos szám	10 (Ah)	A lebegőpontos szám pointerre
Üres lista (mód)	16 (10h)	Közömbös
Szövegkonstans	8 (8h)	A szövegkonstans pointerre
Lista	3 (3h)	A lista első listaelemének pointerre

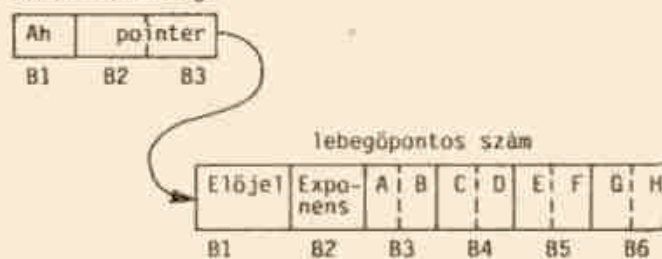
Látható, hogy a binárisan tárolt számok és az üres értéksajton kívüli típusok rendre pointerrel mutatnak értékre. Tekintünk át őket:

Lebegőpontos szám

Hat byte-ot foglal el, az első byte a szám előjelét adja meg, a következő az exponenset, a maradék négy byte-on pedig a nyolc értékes számjegyre van ábrázolva fél byte-onként. Az előjelet az első byte első bitje határozza meg: az 1 a negatív, a 0 a pozitív. Az exponens 128-cal növelt értéke helyezkedik el a második byte-on; programozástechnikai okokból nem megengedett a még ábrázolható legnagyobb kitevő, a +127. (A -128 használható.) A harmadik byte első fél byte-ján van a lebegőpontos szám egész értéke, majd a következő két fél byte-on a tizedesjegyek BCD ábrázolásban.

Sematikusan a következő ábrával jellemezhetjük:

mutató értéksajt



ahol a szám értéke: Előjel . A . BCDEFGH . 10^{Exponens-128}

Az így ábrázolt számok vagy nem egészek, vagy abszolút értékük legalább 32768. Ez a forma a lebegőpontos számok tárolására vonatkozik; amikor műveletet végez velük a rendszer, a fél byte-okon ábrázolt számjegyeket már egész byte-ra terjeszti ki, s egy 11-biteos formában kezeli őket.

Szövegkonstans

Egy értéksajt, majd sorban a karakterei, lezárjuk egy FFh záró byte-tal.

Az értéksajt azt mutatja meg, hogy a szövegkonstans milyen típusú objektumnak a neve (modul, reláció, file...) és hol van az illető objektum részletezése.

Sematikusan:

mutató értéksajt

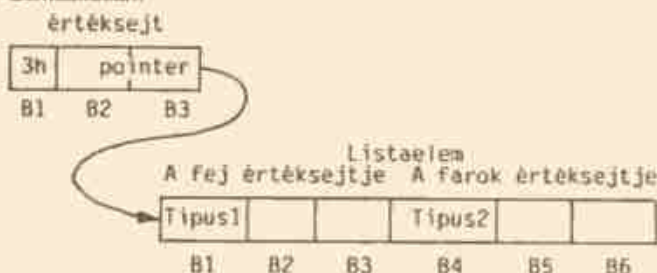


Az n karakterből álló szövegkonstans így n+4 byte-ot foglal el, viszont a programokban többször előforduló szövegkonstansnak csak egyszer kell tárolni, későbbi előfordulásainál a rendszer csak egy rámutató pointer (3 byte-os) értéksajttal generál. Ez a magyarázata annak, hogy a program méretének csak igen csekély növekedésével jár a hosszú, kifejező nevű relációk alkalmazása, akárhányszor is hivatkozunk rájuk.

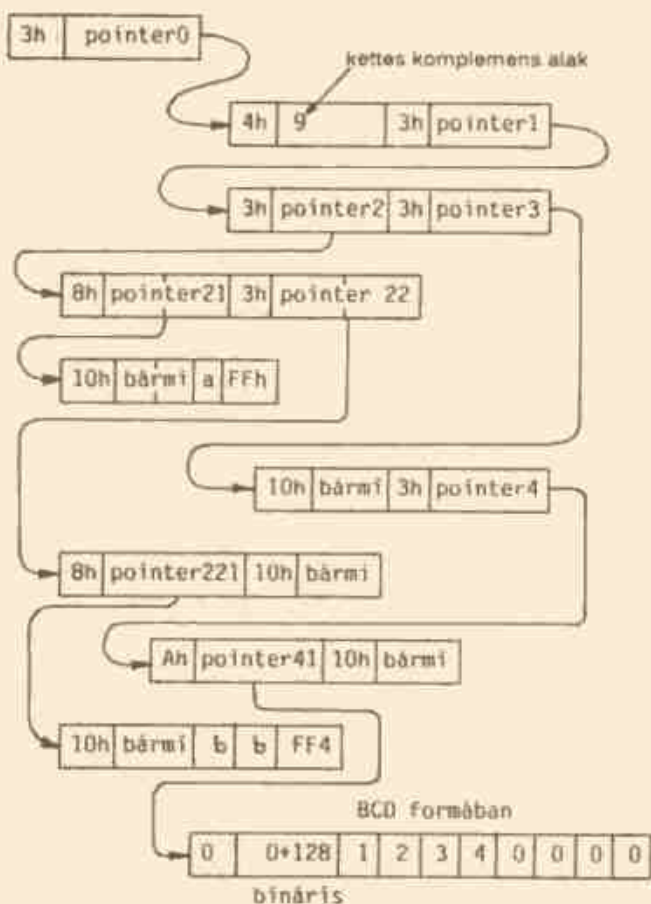
Listaelem

Hat byte-os szerkezet, két értéksejtből álló pár. Az első a lista fejének (vagyis a lista soronlevő elemének) az értéksejtje, a második pedig a lista farkaé. Ez utóbbi általában vagy szintén lista vagy üres értéksejt, ami a lista végét jelzi. Ez az a szerkezeti egység, mellyel tetszőleges módon egymásba ágyazott struktúrákat a legegyszerűbb módon lehet megadni; más nyelvekben (pl. LISP) ezt hívják pontozott párnak.

Sematikusan:



A (9(a bb)) (1.234) lista a következőképpen ábrázolható:



Néhány rendszerrutin használata

Saját gépi kódú relációknak is létre kell hozniuk és kezelniük kell a **micro-PROLOG** adatszerkezeit. Ehhez a legegyszerűbb magának rendszernek a rutinjait használni. Új listaelem vagy lebegőpontos szám létrehozásához a 6593h címen kezdődő rutint kell meghívni, mely lefutása után a HL regiszterpárban megadja egy hat byte-os üres terület címét. Ezt felhasználhatjuk mint listaelemet (két értéksejtként), vagy írhatunk bele lebegőpontos számot is.

Szövegkonstans elhelyezéséhez a legfeljebb 60 byte-os karakter-sorozatot a 9A5h-n kezdődő munkaterületre kell írni FEh-val lezárva, majd a 79E6h-n kezdődő rutint aktivizálva a HL regiszterben megkapjuk a szövegkonstans típusértéksejtjének a címét. A rutin úgy működik, hogy megnezi, létezik-e már ilyen nevű szövegkonstans. Ha van, akkor átmásolja a kupac szabad területé-

re, létrehozza. Ezután a (most már létező) szövegkonstans címét írja a HL regiszterpárba.

A rendszerrutinok működése során – ha már kevés a szabad kupac terület – a hulladékgyűjtő eljárás automatikusan működésbe lép, és minden olyan adatszerkezetet töröl, amely a rendszer számára ismeretlen. Ennek elkerülésére a legegyszerűbb módszer, ha saját adatainkat rögtön beillesztjük egy már létező listába vagy megcímezzük egy adatreigiszter (lásd később) kimenő értéksejtjében. A rendszerben sok más, jól használható rutin van, pl. aritmetikaiak (műveletek és különféle átalakítások), címáncok követése, a rendszerhez tartozás ellenőrzése stb. A gépi kódú kiegészítések száma előtt célszerű átnézni a saját rutin műveleteihez hasonló funkciójú rendszerrutinokat, mert felhasználásukkal sok munkát és helyet lehet megtakarítani! Sok esetben adhat ötletet a rutinok működése a 9800h-9DA4h munkaterület felhasználására, csökkentve a saját adatterület méretét. A további részletek ismertetése meghaladná a munka keretét; a kulturáltan megírt rendszer gépi kódú rutinjainak tanulmányozását minden hozzáértő érdeklődőnek ajánljuk.

Az adatregiszterek

A rendszer a munkaterület 9807h-től 9836h-ig terjedő tartományát 8 darab hat byte-os adatregiszterként használja. Ezeket keresztül bonyolódik a gépi kódú rutinok paraméter átadása. Ez azt is jelenti, hogy legfeljebb nyolc argumentuma lehet egy gépi kódú relációnak.

Az adatregiszterek a következő formájúak:

output			input		
Típus	pointer	pointer	Típus	pointer	Típus
B1	B2	B3	B4	B5	B6

Az első három byte szabályos értéksejt, az a gépi kódú program kimenő (eredmény) értékének a leírására szolgál, belépéskor üres (típusa 10h). A következő három byte a bemenő érték "fordított értéksejtje", azaz a harmadik byte-ban van típusa, s az első kettőben az értéke ill. az értékének a címe.

Az adatregiszterek kimenő értéksejtjébe kell helyezni a rutin eredményének típusát és címét. A visszatérés után ezt akkor veszi figyelembe a rendszer, ha a bemenő argumentum változó típusú volt (csak az elsőn annyit szabad változtatni, ahány argumentuma a relációnak van). Az argumentumok lehetséges típuskombinációit a következő bekezdésben szereplő paraméterfával határozzuk meg. A paraméterfa alapján a rendszer ellenőrzi a bemenő regiszterek típusát, s csak akkor enged tovább a rutint, ha az adott hívás argumentumai megfelelőek, egyébként vagy vezérlési hibát jelez, vagy sikertelen lesz a reláció (vezérlési hibát a változó – nem változó keveredés okoz). Ha nem alkalmazunk paraméterfát, akkor nekünk kell az esetleg szükséges ellenőrzéseket elvégeznünk. Az adatregiszterek bemenő értékét csak olvasni szabad, megváltoztatni nem.

A paraméterfa

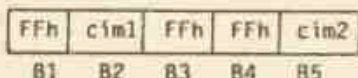
A különböző bemenő argumentum típuskombinációknak egy gépi kódú relációban általában különböző belépési pontok felelnek meg. Ha a **micro-PROLOG** szelleméhez híven pl. azt akarjuk, hogy saját relációnk is többcélú legyen, akkor az összes változatot el kell készítenünk. A megengedett típuskombinációk leírására és a hozzájuk tartozó belépési pontok megadására szolgál a paraméterfa.

A paraméterfát a gépi kódú rutin területén kell felépíteni, 5 byte-os "csúcsokból". Minden byte az adott csúcs kezelőcíméhez képest relatív cím, vagy az FFh érték. A csúcsok első byte-ja a gépi kódú rutin valamelyik belépési pontjának, míg a másik négy byte a paraméterfa többi csúcsának relatív címét tartalmazhatja. E négy byte kitöltése rendre az adott argumentum megengedhető típusát jelzi, megpedig: szám, konstans, lista, változó sorrendben.

Sematikusan:

belépési pont	szám	szöveg konst.	lista	változó
relatív cím	relatív cím	relatív cím	relatív cím	relatív cím
B1	B2	B3	B4	B5

Ha tehát pl. egy reláció legalább egy argumentum és az első argumentum szám vagy változó lehet, akkor a paraméterfa kezdőcíme a következőképpen néz ki:

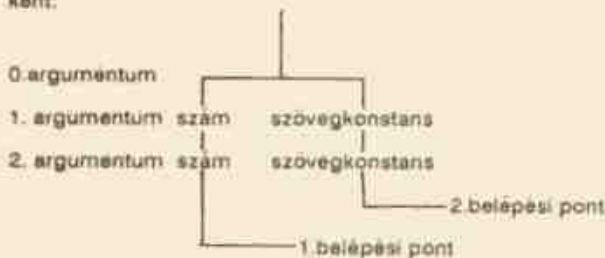


A B1 byte-ban FFh jelzi, hogy argumentum nélküli belépési pontja nincs a rutinnak. A B2-n lévő cím1, ami a B1 byte címéhez képest relatív címe a paraméterfa egy következő számának, azt mutatja, hogy az első argumentum lehet szám és szám esetén a folytatás leírása a megcímzett csúcsnál található. A B3 és B4-en lévő FFh jelzi, hogy az első argumentum sem szövegkonstans, sem lista nem lehet. A B5 byte-ban lévő cím2 (szintén B1 címéhez képest relatív) mutatja, hogy változó vizont lehet az első argumentum és ha valóban az, akkor az itt megcímzett csúcsnál található a lehetséges folytatás. A cím1 és cím2 relatív címeken kezdődő csúcsok ehhez hasonlóan jelzik, hogy második argumentum van-e és típusa mi lehet, illetve mi nem lehet.

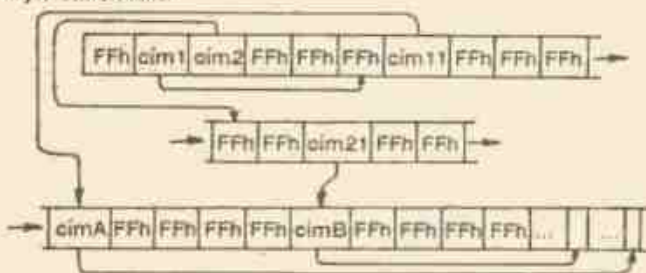
Összefoglalva: A fa kezdőcsúcsából legfeljebb öt él indulhat. Egy a rutin valamelyik belépési pontjához, négy a paraméterfa többi csúcsához. Ha az első byte-on érvényes relatív cím van (nem FFh), akkor ez azt jelenti, hogy a reláció argumentum nélküli meghívásakor erre a címre kell adni a vezérlést. Ha a második byte-on van érvényes relatív cím, akkor a reláció első argumentuma lehet szám és az adott címen folytatódik a paraméterfa. A harmadik, negyedik, ötödik byte hasonlóan jelzi a szövegkonstans, lista és változó típusok elfogadhatóságát. Az első csúcson érvényes relatív címeken lévő csúcsok byte-jal ugyanilyen módon rendelkeznek a második argumentum előfordulásáról és típusáról. A második argumentumhoz tartozó csúcsok érvényes relatív címei adnak információt a harmadik argumentumról, és így tovább. Mivel 8 argumentum lehetséges, elvileg (4-1)/3 csúcsból állhat a paraméterfa. A paraméterfának nem kell grafikai értelemben vett fának lennie, ha a különböző típus kombinációkra azonos feloldozást adhatunk, bizonyos csúcspontok egybeeshetnek.

A paraméterfa alapján történő argumentum ellenőrzést és vezérlést a 65F7h címre adott vezérlésátadással (ugrással, nem hívással) lehet kérni, ha előzőleg a kezdő csúcson címét az IX regiszter-párba töltöttük. A paraméterfa kiértékelő rutin megvizsgálja, hogy az adott hívás argumentumkombinációjához van-e belépési pont cím a paraméterfában. Ha igen, akkor erre a címre adja a vezérlést, ha nem, akkor a reláció kiértékelése sikertelen, vagy vezérlési hibát eredményez.

Nézzünk egy konkrét példát, a LESS reláció paraméterfáját gráf-ként:



Byte-onként kiírt:



Assembly-ben:

CIM0: DEFB -1, CIM1 - CIM0, CIM2 - CIM0, -1, -1
 CIM1: DEFB -1, CIM11 - CIM1, -1, -1, -1
 CIM2: DEFB -1, -1, CIM21 - CIM2, -1, -1
 CIM11: DEFB CIM A - CIM11, -1, -1, -1, -1
 CIM21: DEFB CIM B - CIM21, -1, -1, -1, -1

A paraméterfa ellenőrző rutin hívása Assembly-ben:

```
LD IX, CIM0
JP 65F7h
```

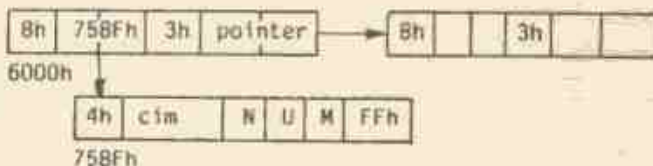
A gépi kódú relációk befejeződése

A paraméterfa kiértékelő rutinja segítségével, vagy anélkül a reláció valamelyik belépési pontjára kerül a vezérlés. Ha sikeresen lefut a reláció, a Z jelzőbit értékét be kell állítani, az esetleges kimenő argumentumokat a micro-PROLOG konverzióknak megfelelően (lásd a Szintaktikai egységek ismertetésénél) el kell készíteni, és az értékejtüket a megfelelő adatregiszter első három byte-jába be kell írni. Sikertelen esetben törölni kell a Z jelzőbitet. A micro-PROLOG rendszernek a verem tetején lévő cím felhasználásával (pl. RET) kell visszaadni a vezérlést.

A reláció rendszerbe illesztése

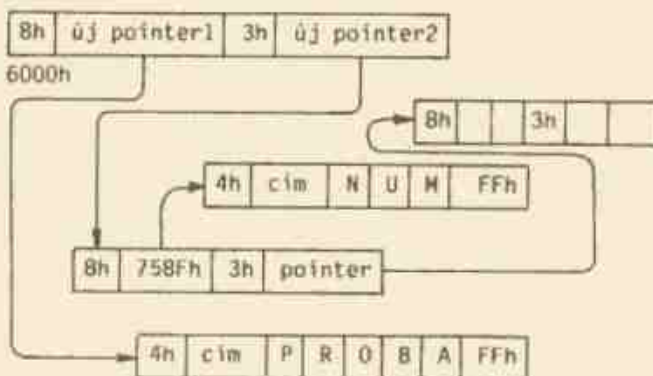
A reláció nevét szövegkonstansként el kell készíteni, és be kell illeszteni a rendszer eljárásait tartalmazó listába. A szövegkonstans értékejtje 4h típusú, értéke a gépi kódú rutin kezdőcíme. A rendszer eljárásait tartalmazó lista értékejtje 6000h-n van, ebbe a listába kell beszúrni az új eljárást. Az a legegyszerűbb, ha egy olyan listaelemet készítünk, amelynek második fele megegyezik a 6000h-n található értékejtjével, az első fele pedig a reláció előzőleg ismertett szövegkonstansára mutat. A 6000h-n lévő listaelem első értékejtjének mutatóját az általuk készített listaelem címére kell változtatni.

Sematikusan:



A PROBA nevű gépi kódú rutint az első rendszerreláció, a NUM elé kell illeszteni

Beillesztve az új relációt:



Megjegyzések

- A micro-PROLOG rendszerrelációk kétféle típusúak, gépi kódúak vagy micro-PROLOG-ban megírtak. A micro-PROLOG-ban írtak ugyanolyan formájúak, mint a felhasználó által micro-PROLOG-ban készített relációk (csak nem törölhetőek). Ilyen jellegű kiegészítést is készíthetünk Monitor/Assembler programok segítségével. Ehhez a feladathoz el kell készíteni a PROLOG nyelvű relációt listaként (lásd: Szintaktikus egységek), majd ennek a listának a szövegkonstansát kell a rendszer rutinok listájába illeszteni. A módszer ugyanaz, mint a gépi kódú rutinok esetében, azaz a kivétel, hogy a szövegkonstans típusa 3h, s pointerre a reláció listájára mutat. Ne feledkezzünk meg a 966h és a 7657h címen elhelyezkedő, a kupac terület elejére (9DA5h) mutató értékeknek a programunk által elfoglalt mérettel való megnöveléséről!
- A gépi kódú program megírása és beillesztése bármely Monitor/Assembler segítségével elvégezhető, mi a GENS/MONS programokat használtuk.

HISOFT C

Adattípusok használata

A C nyelv alap-típusairól már szöveztünk. Ebben a részben arra a nem éppen egyszerű kérdésre keresünk választ, hogy a C-ben hogyan lehet származtatott típusokat definiálni. Származtatott típus a következők valamelyike: **mutató**, **tömb**, **visszatérési érték**, **struktúra**, **unio**. Mindezeket rekurzív módon definiálhatjuk, tehát lehet olyan struktúránk, melynek egyik tagja egy tömb, másik pedig egy ugyanolyan típusú struktúrát megcímző mutató. Nézzük az egyes adattípusok definiálási módjait!

Mutatók: A mutatók használata nélkülözhetetlen. Egyrészt egyes adattípusok csak a segítségükkel adhatók át függvények argumentumként, másrészt a mutatók segítségével lehet bonyolult adattípusokat definiálni. Mutatót létrehozni a típus * deklarátor utasítással lehet, ennek hatására a deklarátor által meghatározott azonosító egy "típus" típusú változót megcímző azonosító lesz. Néhány egyszerű példa.

- char *p, *q;
- unsigned *szamialo;
- static char *base;

Az első sor két p és q nevű mutatót deklaráli, amelyek char típusú változót címeznek meg. A második példa egy **szamialo** nevű mutatót határoz meg, amelyik egy előjel nélküli egész számot címez meg. Végül a harmadik példa mutatja, hogy tárolási osztályt is rendelhetünk a deklarációk elé, a **base** nevű char típusú mutató a deklarációt tartalmazó függvény első hívásakor létrejön és értéke a program futása ideje alatt végig megmarad. A másik két példa változó értékei a függvényekbe való belepéskor mindig definiáltak.

A mutatókkal aránylag kevés művelet végezhető. Egyik ilyen az értékadás. Ha p és q azonos típusra mutatnak, úgy mint az első példánkban, akkor a p = q utasítás hatására p ugyanazt a memóriaterületet fogja megcímezni, mint q. A következő művelet a mutatók összeadása és kivonása. Ennek megértésére tekintsük a **szamialo + 1** értékadást, amelyben **szamialo** előjel nélküli egészre mutat, **szamialo + 1** a következő előjel nélküli egész számot címez meg. Mivel a gépi megvalósításban ez 2 byte, ezért a **szamialo + 1** a memóriának 2-vel nagyobb címére fog mutatni. Természetesen mutatókra használhatjuk a p + + vagy p + - n típusú értékadásokat is.

Ez a tulajdonság tetszőleges mutatókkal végzett művelet során megmarad, ha a mutató által megcímzett típus tárolására n byte szükséges, akkor a mutató 1-gyel való növelése vagy csökkentése a mutató valójában n memóriahellyel mozgatja előre vagy hátra.

Lehetőség van azonos típusú mutatók összehasonlítására: p < q; p > q; p < = q; p > = q. Ezek jelentése a várt, pl. p < q azt jelenti, hogy a p által megcímzett memóriarész előbb kezdődik, mint a q által megcímzett. Nagyon hasznos a

cast (típusnév) kifejezés;

forma használata. Ennek hatására a kifejezésből, amely tetszőleges típusú kifejezés lehet, a rendszer "típusnév" típusú változót készít. Ennek a mutatók esetében rendkívül fontos szerepe van!

A leggyakoribb eset azonban az, amikor a mutatók által megcímzett értékekkel végzünk műveletet. Tekintsük pl. a következő deklarációt:

```
int *p, *q, x;
```

Ekkor a *p = x, *p = *q és az x = *p értékadások megengedettek. Hatásuk rendre a következő:

- 1) a p által megcímzett memóriaterületre másolódik x értéke.
- 2) a q által megcímzett egész értéke átmásolódik a p által megcímzett memóriarészbe, végül
- 3) a p által megcímzett memóriarészben lévő egész átmásolódik az x egész számba.

Mutatók kezdő értéke: A mutató deklarálásakor a C fordító csak a mutatónak foglal helyet, a mutató által megcímzett típusnak nem. Azt a tényt, hogy a mutató még nem mutat értelmes helyre, úgy szokták kifejezni, hogy a mutató értéke NULL. A gépi reprezentációk ezt általában úgy interpretálják, hogy a NULL mutató által megcímzett memóriát éppen a 0-as cím, mert ez által-

ban sohasem használható. A C fordító minden mutatót NULL-nak inicializál, kivéve ha expliciten mást nem mondunk.

Ebből következik, hogy az előző bekezdésben látható példák önmagukban mind értelmetlenek, hiszen a mutatók által megcímzett egészeknek még nem foglaltunk helyet a memóriában. A memóriafoglalást kétféleképpen tehetjük meg: vagy egy már létező (statikusan deklarált) változóra mutatunk, vagy a program által előre le nem foglalt memóriából kérünk helyet. Az előbbi az egyszerűbb. Tegyük fel pl., hogy s és t valahol már deklarált egész szám. Ebben az esetben a

```
p = &s; q = &t;
```

értékadásokkal a p és q mutatót úgy állíthatjuk be, hogy az általuk megcímzett memóriarész éppen az s és a t változók által elfoglalt memóriarész legyen. Ezután a *p = x értékadás már nem hiba, hiszen a program már tudja, hogy a p által megcímzett memóriarész micsoda.

A C (és általában a magasszintű nyelvek) az operációs rendszertől kérnek egy tárterületet, amit **HEAP**-nek hívnak. Ez arra szolgál, hogy azoknak a mutatóknak is (pontosabban az általuk megcímzett objektumoknak) helyet tudjunk biztosítani, amelyek nem egy előre deklarált statikus változóba akarják tárolni az általuk mutatót értékeket. Ezt hívjuk **dinamikus memóriának**, mert nem lehet előre tudni, hogy a **heap** hogyan lesz felhasználva. Tekintettel arra, hogy Spectrum-ban nincs operációs rendszer, célszerű lenne **heap**-nek a fennmaradó memóriarészt teljes egészében lefoglalni. Kár, hogy ennek a méretét előre nem tudjuk... A **HISOFT C** fordító ezért pontosan **1000 byte-ot foglal le heap-nek**. Hogy hogyan, arról a későbbiekben még részletesen szöveglünk. Egyelőre elégedjünk meg azzal, hogy a memóriafoglalás a **HISOFT C** esetén a

```
char *calloc(n,meret)
```

függvény meghívásával történik, ami egy olyan mutatóval tér vissza, amelyik n darab, egyenként **meret** hosszú objektumot címez meg. Ha például a p és q változóinknak helyet akarunk foglalni, akkor azt a

```
p = cast(int) calloc(2,sizeof(int)); q = p + 1;
```

utasításokkal tehetjük meg. Az első értékadás foglalja le a memóriában a helyet. Ez két memóriablokk, melyek egyenként **sizeof(int)** nagyságúak. Ez természetesen összesen 2*2 helyet jelent, de a fenti felírási mód gépfüggetlen. Végül a q = p + 1 értékadás a p mutató értékét 2-vel (az int tárolási hosszával) növeli meg.

Tömbök: Kezdjük a **isgnehezbe**bel: a C-ben minden egyes tömb egy konstans mutató. Ez a mutató a tömb elemszámától függő, a memóriában egymás után elhelyezett azonos típusú elemek közül az elsőre mutat. Most próbáljuk ezt közérthetően almon-dani. Tömböt létrehozni az alábbi deklarációk egyikével lehet:

- típus deklarátor[kifejezés];
- típus deklarátor[];
- típus deklarátor[] = kezdőérték;

A fenti definíciók hatására a deklarátorban szereplő azonosító egy tömb lesz, melynek kifejezés darabszámú eleme van, amelyik mindegyikének "típus" a típusa. A második esetben a darabszámot valahonnan máshonnan tudja meg a rendszer, pl. kezdőértéket adunk neki, ahogy azt a harmadik lehetőség mutatja.

Tegyük fel, hogy a deklarátor az "azonosító" nevű változót deklarálja. Ekkor az első tömbdeklaráció – definíció szerint – a következővel ekvivalens:

```
típus * deklarátor;
```

```
azonosító = calloc(kifejezés,sizeof(típus));
```

A második esetben a memóriafoglalásra nem kerül sor, míg a harmadik esetben a tömb elemeinek a számát a kezdőértékből állapítja meg a fordító, s annak megfelelően hajtja végre a helyfoglalást a rendszer. Ezt a memóriarészt azonban nem a **heap**-en foglalja le a rendszer, hanem statikusan beleépíti a programba.

Tömbökkel végezhető műveletek: Tekintettel arra, hogy a tömbök valójában mutatók, pontosan ugyanazok a műveletek végezhetők el velük, mint a mutatókkal. Kivételt képez a tömbelemek indexelése. Ha "azonosító" egy tömb, akkor ennek a tömbnek az i. elemére az **azonosító[i]** formában hivatkozhatunk. Mutatók használatával ez az *(azonosító+i) hivatkozással írható le.

Nézzünk néhány egyszerű példát tömbökre:

```
int x[5], y[3][5], z[] = {1, 2, 3}, u[];
```

A fenti deklaráció hatására sorra a következők jönnek létre: *x* nevű 5 elemű tömb, *y* nevű tömb, amelyik 3*5 egésznek foglal helyet, *y* valójában három elemű tömb, melynek minden eleme öt egész számot tartalmazó tömb. Ha a többdimenziós tömböket mátrixnak képzeljük, akkor az elemek tárolása sorfolytonos, *z* három elemű tömb lesz, melynek értékei sorra 1, 2 és 3. A fordító a kezdőérték alapján számítja ki a *z* tömb nagyságát. Az `u[]` deklaráció csak függvény formális paramétereinek a megadásában szerepelhet. Az *u* tömb számára a fordító nem foglal külön helyet, az *u* értéke a függvény hívásakor átadott tömb lesz, annak nagysága magából az *u*-ból nem számítható ki. Az üres `[]` használata csak egyszer megengedett, az `u[][]` forma szintaktikus hibát eredményez.

A fent deklarált változók segítségével számos hivatkozás lehetséges. A legegyszerűbbek a tömbök elemeire való egyszerű hivatkozások: `x[2]`, `y[1][3]`, `z[2]`, `u[1]`. A többdimenziós tömbök esetében nem megengedett az `y[1,3]` alak használata! Az *y* tömbnek azonban valamely sorára is hivatkozhatunk, pl `y[2]` a tömb második sorát jelenti, amelyik egy 5 elemű tömb. Így az `y[2]=x` értékadás megengedett, *s* hatására az *x* tömb összes eleme átmásolódik az *y* tömb első sorába!

Van azonban a tömbnév és a mutatók közt egy lényeges különbség: a tömbnév mint mutató konstans. Ez azt jelenti, hogy tömb=*dx* alálú értékadás nem lehetséges. A program életében a tömb mindig a memóriának ugyanarra a helyére mutat!

Karakterláncok (sztringek) mint tömbök avagy mint mutatók: A C nyelv további sajátossága, hogy a karakterláncokat speciális mutatókként kezeli. Ez pontosabban azt jelenti, hogy a karakterlánc karakterek sorozata, amelyik egy 0 byte-tal végződik. A 0 byte-ra a C-ben a `'\0'` formában szokás hivatkozni. A karakterláncok végére a 0 byte általában automatikusan odakerül. Például a

```
char * karakterlanc = "abcdef",
```

deklaráció hatására a karakterlánc mutató egy olyan memóriaterület elejére fog mutatni, amelyik egy "abcdef" karakterláncot és a végén egy 0 byte-ot tartalmaz. Szerencsére a sztringműveletek is automatikusan érzékelik a 0 byte-ot, *s* nem nekünk kell vele foglalkoznunk. A mutatók és a sztringek fenti azonosítására álljon itt példaként a sztring hosszát kiszámító függvény:

```
int strlen(s) /* Visszadja s hosszát! */
char *s;
{
    char *p = s;
    while (*p) p++;
    return (p - s);
}
```

Nos, az első két sor azt definiálja, hogy `strlen(s)` egész értéket visszaadó függvény, melynek egyetlen argumentuma egy karaktermutató. A függvény teste egy újabb deklarációval kezdődik, amelyik a *p* `char` típusú mutatót deklarálja és rögtön inicializálja is az eredeti sztring első karakterére. A `while` ciklus feltételeiben `*p` szerepel. Mint tudjuk a `while` akkor hajtódik végre, ha a feltétel igaz, azaz az értéke nem 0. Így a ciklus addig jár, míg a *p*-vel megcímezett byte 0 nem lesz. A ciklusmag nem csinál mást mint folyamatosan növeli a *p* mutatót. Végül a `return` utasításban a két mutató különbsége pontosan a nem 0 byte-ok számát, azaz a sztring hosszát, adja vissza. (Himmm... jó ha valami tömör, de ennyirel)

Már az eddigiek is mutatják, hogy a mutatók, még olyan esetekben is mikor nem kellene, rendkívül fontos szerephez jutnak. A C-ben végül is minden a mutatókra vezetődik vissza. Mutatja ezt az is, hogy tetszőleges *p* mutatóra használhatjuk a `p[i]` formát, függetlenül attól, hogy a *p* tömbnek volt-e definiálva vagy sem!

A mutatók további fontossága, hogy a függvények argumentumai nem lehetnek **struktúrák, uniók és függvények** csak ilyeneket megcímező mutatók!

Visszatérési értékek: A C programokban lehetőség van függvények önálló deklarálására, *s* arra hogy a függvény definícióját valahol máshol adjuk meg. Hasonlóképpen lehetőség van függvényeket megcímező mutatók aktuális paraméterként való átadására. Ilyen esetben azt szoktuk mondani, hogy a szóbanforgó azo-

nosító visszatérési értéket deklarált. Ennek formája a következő:

```
tipus deklarátor()
```

A deklarátorban lévő azonosító függvény, amelyik "típus"nak megfelelő értéket ad vissza. Ebben a formában a függvény argumentumainak típusát nem kell megadni, bár lehet. Tekintsük a következő példákat:

```
• int strlen();
• int strlen(char *);
```

Az első példa megad egy `strlen` nevű függvényt. A második ugyanazt adja meg, de tudjuk róla, hogy az első argumentuma egy `char` típusú megcímező mutató. A HISOFT C nem engedi meg a második fajta deklaráció használatát. (Ezt egyébként absztrakt típusdeklarációnak hívjuk, tekintettel arra, hogy a típusdeklarációban nem szerepel változóneve.)

Struktúrák és uniók: A C utolsó típusmegadási lehetőségéhez értünk el, amelyik egyben lehetővé teszi a fentiek igen rugalmas kihasználását. A **struktúrák** és az **uniók** szintaxisa nagyon hasonló (mi több azonos) ezért egyszerre lehet ezeket a deklarációkat fajtáknak leírni. A **struktúra** és **unió** deklarációk alakja a következő:

```
struct { < struktúra deklaráció lista > } deklarátor;
struct azonosító { < struktúra deklaráció lista > } deklarátor;
struct azonosító deklarátor;

union { < struktúra deklaráció lista > } deklarátor;
union azonosító { < struktúra deklaráció lista > } deklarátor;
union azonosító deklarátor;
```

A **< struktúra deklaráció lista >** deklarációk tetszőleges sorozata lehet. A **struktúra** a listán felsorolt elemekből tevődik össze, míg az **unió** olyan változó, amelyik a listán felsorolt objektumok bármelyikének a tárolására képes. A struktúrákkal és az uniókkal természetüknél fogva eltérő műveletek végezhetők. Az unióval minden olyan művelet elvégezhető, amelyik valamelyik tagjával. A fordító nem ellenőrzi, hogy ez a művelet amúgy értelmes-e. A struktúrával önmagával a másoláson (azaz a *p=q* és hasonló értékadásokon kívül) más művelet nem végezhető. Végezhetőek azonban műveletek a struktúra tagjaival. Ha a struktúra valamelyik tagjának a neve "resz", a struktúra által deklarált azonosító pedig "fo", akkor `fo.resz` a `fo` struktúra "resz"-nek megfelelő tagját jelenti, *s* ez minden olyan kifejezésben szerepelhet, ahol a "resz"-nek megfelelő típusú értékek.

Nézzünk néhány példát a struktúrákra és uniókra:

```
struct racionalis {int O,N;} racionalis;
struct datum {int ev; byte ho, nap;} mainap;
union {char *s; int i;} miez;
```

Az első példa egész számpárokat deklarált. A struktúra neve "racionalis", *s* egyben definiáltunk egy struktúra változót, aminek a neve ugyancsak "racionalis". A struktúra két tagjára a `racionalis.O` és a `racionalis.N` jelölésmóddal hivatkozhatunk, ezek egész számokat jelentenek. A második példa a `datum` nevű struktúrát definiálja, amelyik egy egész számból és két byte-ból áll, *s* ezekkel rendre az évet, a hónapot és a napot szeretnénk leírni. A deklaráció a `mainap` nevű változót deklarálja. Ha történetesen 2001 január 35-ét írunk, akkor ezt a

```
mainap.ev = 2001; mainap.honap = 1; mainap.nap = 35;
```

értékadásokkal fejezhetjük ki. Tekintettel arra, hogy a struktúrák megadtuk a nevét is (`datum`) ezért a későbbiekben már használhatjuk a

```
struct datum hatarido
```

megoldást is, melynek hatására egy új `datum` típusú struktúra-változó jön létre `hatarido` névvel.

Az utolsó példánk az **uniók** használatát szemlélteti. A `miez` nevű változónak összesen 2 byte-nyi hely foglaldók, mert a mutatónak `ie` és az egésznek is ennyi tánterületre van szüksége. Ha a `miez`-t mutatóként akarjuk használni, akkor `miez.s` alakban kell rá hivatkoznunk. Ha viszont azt akarjuk, hogy a rendszer egészként számoljon vele, akkor a `miez.i` formában hivatkozhatunk csak rá.

Mutatók és struktúrák: Előfordulhat, hogy egy mutató egy struktúrát címaz meg. Ez az eset annyira gyakori, hogy külön jelölést vezettek be a megcímezett struktúra tagjához való hozzáférésre. Legyen *p* egy struktúrát megcímező mutató, `resz` pedig a struktúra egyik tagjának a neve. Ekkor a `p->resz` jelölés ekvivalens a `(*p).resz` használatával. Ugyanez igaz az uniókra is.

A struktúrát megcímező mutatók használata azért gyakori, mert

segítségével lehet rekurzív adattípusokat definiálni. Általában semmilyen programozási nyelvben sem megengedett, hogy egy struktúra definíciójában maga a struktúra újból szerepeljen, de szerepelhet a struktúrát megcímző mutató. Egy háromelemű tömbökből álló listát pl. a következőképpen lehet definiálni:

```
struct peldarek {int x[3]; struct peldarek *kovetkezo;} lista;
Ez a struktúra definíció egy "peldarek" nevű struktúrát definiál, melynek két tagja van: egy háromelemű tömb, s egy peldarek struktúrát megcímző mutató. A lista nevű változó 8 byte-nyi helyet foglal el, s néhány lehetséges értékadás a lista változóra a következő:
```

```
lista.x[2] = 5;
lista.koetkezo = NULL;
lista.koetkezo.x[2] = 7;
```

Megjegyezzük, hogy az utolsó értékadásnak csak akkor van értelme, ha lista.koetkezo a NULL-tól különbözik.

A typedef használata: Gyakran van szükség arra, hogy bizonyos adatszerkezeteket önmagukban definiáljunk, s ezután használjuk ezeket bizonyos adattípusok megnevezésére. A typedef maga tehát nem hoz létre semmilyen új adattípust, egyszerűen kényelmesebbé teszi a bonyolult adattípusokra való hivatkozást. Használata egyszerű:

```
typedef típus deklarátor;
```

A typedef szócska nélkül a fenti utasítás a deklarátorban lévő azonosítót deklarálná a deklarátorban megadott módon. A typedef használatával azonban annyi történik, hogy az azonosító ezután a fenti (typedef nélküli) deklaráció rövidítésére szolgál. Nézzünk két egyszerű példát:

```
typedef int *muttomb[100];
muttomb A,B,C;
```

A typedef definiálja a muttomb nevű típusnevet, amelyik ezután bárhol használható. Hatása ugyanaz, mintha az A,B és C neveket a muttomb helyére tettük volna a megfelelő typedef után álló deklarációba:

```
int *A[100];
int *B[100];
int *C[100];
```

A típusképző műveletek iterálása Már a fenti példák is illusztrálják, hogy a felsorolt típusképző utasítások tetszőleges mélységben egymásba skatulyázhatóak s akár rekurzív módon is használhatóak. Nagyon gondoljuk azonban meg, hogy milyen adattípusokat is használunk, hiszen a mutatók szerepe miatt könnyen felülírhatjuk saját programunkat is! Néhány egyszerű példa az összetett típusképzésre:

```
int (*tomb)[20];
int *tomb[20];
char **a;
```

Ezzel magának a C nyelv felépítésének az ismertetését befejeztük. A következőkben ismertetni fogjuk a HISOFT C memóriakezelését, a beépített és Spectrum függő függvényeket s természetesen sok-sok példát fogunk adni a C nyelv használatára.

Most egy egyszerű példát ismertetünk, amelyik a strukturák használatát szemlélteti. Mint említettük, a HISOFT C nem tartalmazza a valós számokat, csak az egészeket. Ezen segít az alábbi program, amelyik törtszámokkal képes műveleteket végezni. Íme a program:

```
#define void int
typedef struct { int N,Q; } rac;
```

```
/* Vigyázat: ez az igazi C definíció,
   azonban itt nem így kell megadni!
rac elso = {1,12}, masodik = {5,12}; */
rac elso = {1,0,12,0}, masodik = {5,0,12,0};
```

```
rac *norm(a)
rac *a;
{
  int legoszt;
  rac eredmeny;
```

```
if (a->O == 0) {eredmeny.N=0; eredmeny.O=0;
return(&eredmeny);}
if (a->N == 0) {eredmeny.N=0; eredmeny.O=1;
return(&eredmeny);}
```

```
legoszt = inko(a->N,a->O);
```

```
eredmeny.N = a->N / legoszt;
eredmeny.O = a->O / legoszt;
```

```
if (a->O == 0) {eredmeny.O = -eredmeny.O; eredmeny.N
= -eredmeny.N;}
```

```
return(&eredmeny);
```

```
}
```

```
int inko(u,v)
```

```
int u,v;
```

```
{
```

```
int r;
```

```
if (u < 0) u = -u;
```

```
if (v < 0) v = -v;
```

```
if (u < v) ar=v; v=u; u=r;é
```

```
r = u % v;
```

```
while (r != 0)
```

```
{
```

```
u = v;
```

```
v = r;
```

```
r = u % v;
```

```
}
```

```
return(v);
```

```
}
```

```
void kiir(a)
```

```
rac *a;
```

```
{
```

```
if (a->O == 1) printf("%d",a->N);
```

```
else
```

```
printf("%d /%d",a->N,a->O);
```

```
}
```

```
rac *osszeg(a,b)
```

```
rac *a, *b;
```

```
{
```

```
rac eredmeny;
```

```
eredmeny.N = a->N * b->O + b->N * a->O;
```

```
eredmeny.O = a->O * b->O;
```

```
return(norm(&eredmeny));
```

```
}
```

```
main()
```

```
{
```

```
kiir(osszeg(&elso,&masodik));
```

```
}
```

A program elején két egész számból álló párként definiáljuk a tört számokat (amiket hagyományosan racionális számoknak hívnak, innen a nevük: rac. Az N a számláló, az O a nevező, azaz a szám maga N/O. A legérdekesebb a norm függvény, ez előállítja egy racionális szám ún. normál alakját. Pl. az 8/-12 számból -1/2 lesz.

A további függvények a számok összeadását és kiírását valósítják meg. A norm függvényhez szükséges még egy segédfüggvény is, ez az inko függvény, amelyik két, nullától különböző egész szám legnagyobb közös osztóját számítja ki.

Figyeljünk meg, hogy mindenütt ahol a függvények argumentuma vagy a visszaadott függvényérték rac típusú ott mutatót használunk a strukturák helyett (mert a nyelv csak erre biztosít lehetőséget...).

Bevezetés

A program négy alapvető szolgáltatása a következő:

1. Elolvassa a "fejléceket" kívül a kódolt információt, ami a fejlécben található.
2. Megadja a blokkhosszt (ha a fej: 17 byte-ot tartalmazott)
3. BYTES REMOVER (byte-irtó) opció, amely a programok kezdetétől számított legfeljebb 16380 byte eltávolítását teszi lehetővé (fejlel vagy fej nélkül), majd azt kódként kimentve azután.
4. Saját készítésű 17 byte-os fejléc kimentését teszi lehetővé.

A program indítása

1. A LOAD™ kiadása után a program automatikusan betöltődik.
2. A betöltés után az alábbiakban részletezett négy opció közül választhatunk. A BREAK-et bármikor használva a főmenübe térünk vissza.

1. OPCIO: fejelelvás

- a) Csévéljük vissza az elejére azt a szalagrészt, ahol a fejlécet "el akarjuk olvasni", úgyelve arra, hogy elegendő hosszú legyen a TONE is rajta.
- b) Nyomjuk meg az '1' billentyűt és indítsuk el a magnót.
- c) Amikor a fejléc beolvasódott, az opció az alábbi információkat közli:
 - a fejlécben lévő nevet
 - a program típusát
 - a program hosszát
 - ha kód-típusú, akkor a töltési címet,
 - ha BASIC típusú, akkor az indítósort,
 - ha tömb, akkor annak betűjelét
 - a BASIC rész önálló hosszát.
- d) Az opció az '1' billentyű ismételt megnyomásával lehet újra működésbe hozni. A főmenühöz a '2' billentyű segítségével térhetünk vissza.

2. OPCIO: byte számláló

Ez az opció bármely kódblokk hosszát meg tudja adni.

- a) Tekerjük a szalagot annak a résznek az elejére, amelynek hosszára kíváncsiak vagyunk.
- b) Nyomjuk meg a '2' billentyűt és indítsuk el a fejlelvást, melynek végén kirodók, hogy hány byte-ból állt a blokk. Gyakran jó ezt tudni pl. a hamis fejek esetében.
- c) Ha újra igénybe akarjuk venni ezt az opciót, itt is az '1' tud billentyűvel ismételtethetünk, és a '2' billentyű visz vissza a főmenühöz.

3. OPCIO: fejgyártó

Ez az opció egy tetszés szerinti, 17 byte-os fejléceket képes kimenteni a szalagra:

- a) A főmenüből a '3' billentyű megnyomásával juthatunk ebbe az opcióba, amely azzal jelentkezik be, hogy a fejlécben tárolni kívánt 10 karakteres névre rákérdez: "Program name?" Ugyeljünk, mert 'ENTER' közvetlen megnyomására a főmenübe kerülünk vissza!
- b) A név megadása után kapott kérdés: "BASIC or CODE?" Numerikus vagy karakter tömbökhöz nem készíthető fej!
- c) Most a blokk hosszára vonatkozó kérdésre kell válaszolnunk: "Length in bytes?"
- d) Ha a fejlel BASIC típusúnak választottuk, akkor a következő kérdés az automatikus indítás sorának számát szeretné meg tudni: "Automatic line number?", amely sortól betöltéskor a programnak magától kell elindulnia. Ha erre a kérdésre 32767-nél nagyobb számot adunk meg, a program nem lesz automatikus indítású, ez esetben RUN-nal kell indítani! Ha CODE típusút választottunk, akkor a kérdés arra vonatkozik, hogy mi legyen a kód betöltési címe. Tudjuk, hogy önmagában ez a betöltési cím nem kényszerítő erejű, hiszen megadható bármilyen kód-jellelű blokknál, hogy hová kívánjuk rakni, mindössze azt szolgálja, hogy a kód akkor is tudja, hová kell betöltődnie, amikor csak LOAD™ CODE utasítást adunk ki.
- e) Az utolsó kérdés ("Length of BASIC?") a BASIC rész hosszát firtatja. Legcélszerűbb, ha a kód esetében mindig a kód tényleges hosszát adjuk meg!

4. OPCIO: byte töröl

Némelyik program egész egyszerűen túl hosszú ahhoz, hogy a microdrive megbirkózzék vele, ezért vagy meg kell rövidíteni, vagy szét kell hasítani, esetleg mindkét műveletre szükség van. Ennek az opciónak a segítségével maximum 16380 byte hosszú részt lehet lehasítani egy BASIC, CODE, vagy HEADERLESS (fejlel) byte-blokk elejétől és kódként szalagra menteni, ami a nyelés utáni MARADÉK.

- a) A főmenüből a '4' megnyomásával választjuk ki ezt az opciót, amely azzal a kérdéssel jelentkezik be ("header information already stored?"), hogy van-e előkészített és elraktározott információ a fejrész számára. Hogy miért? Tegyük fel, valamilyen BASIC vagy CODE típusú program méretét kívánjuk csökkenteni. A fejrész elolvasására az '1' opciót használtuk fel. Ha ez így van, akkor a programnak íymódon valóban van már neve stb, tehát takarékoskodva az idővel megnyomhatjuk az 'Y' billentyűt igenítő válaszként. Amennyiben itt 'N'-t nyomunk, akkor a 3. fejjgyártó opcióhoz igen hasonló fázisokon kell keresztülnünk, azzal a különbséggel, hogy most nem a program hosszát, hanem az eltávolítani kívánt byte-ok számát kell megadnunk.
- b) A fejkészítés lépésain túlcsve megjelenik a "PLAY THE TAPE" üzenet. Amennyiben a program BASIC vagy CODE típusú, úgy a betöltést a fejlécben meghatározott címtől kell behívni. Természetesen, ha fejlel, akkor a fejlel rész fizikal kezdetétől kell indítanunk a betöltést.
- c) Ne rémüljünk meg a képernyőn – a betöltés alatt – megjelenő furcsa ábráktól. Amikor a betöltés befejeződött, meg lehet kezdeni a megjelölt rész kimentését: ezt 'ENTER' indítja. NE GYANAKODJUNK HIBÁRA, amikor a mentés befejeztével a program NEW parancsot ad ki, és kiírja önmagát.

Beszéeljünk inkább a szolgáltatás korlátairól:

- Pontosan csak az utolsó 48700 byte-ot képes kimenteni, tehát a legtöbb program esetében elboldogul a kívánt hosszakkal.
- Mit lehet tenni pl. egy 60000 byte hosszú programmal?
- Minthogy kimenteni csak az utolsó 48700 byte hosszú szakaszt lehet, ki kell itani legalább 60000-48700=11300 byte-ot, hogy a program hibátlanul betölthesse az utolsó byte-ot is. Ez nem jelenthet valódi problémát, hiszen a program tényleges működéséhez legfeljebb 41983 (65535-23552) byte mindenképpen elegendő.
- Jegyezzük meg, hogy van, amikor kétszer is sor kerül ennek az opciónak a használatára: pl.
 - először egy 41000 byte kiemeelésére,
 - majd ahhoz, hogy ezt a 41000-es blokkot két különböző nagyságú részre lehessen bontani, mondjuk egy 3000 és egy 38000 byte-nyi szakaszra. Ennél az utóbbinál kell a '4' opciót használni, az első 3000-hez pedig az 'INBYTES' kódhasítót.

GYAKORLAT: Tetszőleges programok esetén próbáljuk ki a 2. OPCIO-t az egyes szakaszok hosszának megállapításához. Ellenőrizzük, megfelelnek-e a tényleges hosszak a fejlécben jelzeteknek. Ha nem, akkor annál a programnál "hamis" fejlel alkalmaztak a gyártók.

Hozzuk a Spectrumot alapállapotba, majd írjuk be és indítsuk el a következő programot, mely kódot ment szalagra:

```
10 CLEAR 24999: FOR a = 1 TO 200: POKE (A + 24999), a:
NEXT a 20 SAVE "200 byte" CODE 25000,200
```

A megjelölt helytől (25000, 25199) kezdve az 1...200 számok írórak be a memóriába, majd mentődnek ki szalagra. Miután kimentettük szalagra ezt a 200 byte kódot, hívjuk be a fejelelvásó programot. Olvassuk el ennek az általunk "200 byte"-nak elkeresztelt kódnak a szalagon talált fejlel az 1.OPCIO-val. Vegyük igénybe a 4.OPCIO-t a 200 byte első 50 byte-jának eltávolítására. Az "Is the header prepared?" kérdésre nyugodtan felelhetünk 'Y'-t, hiszen már használtuk az 1.OPCIO-t! A most már csak 150 byte-nyi rész kimentése után ellenőrizzük, hogy minden rendben van-e:

Adjuk ki a RANDOMIZEUSR 0 utasítást a memória kitakarítására, majd hajtsunk végre egy CLEAR 24999 parancsot. Hívjuk be az új kódot a LOAD™ CODE 25000 révén, ezután gépeljük be ezt az egysoros utasítást:

```
10 FOR a = 25140 TO 25160: PRINT a, PEEK a: NEXT a
Futtassuk most ezt a programot. Azt kell észlelnünk, hogy a 25149 cím után már minden rekesz zárult tartalmaz.
```

Ami igaz az igaz, az LSI kiadásában megjelent SINCLAIR SPECTRUM JÁTÉK ÉS PROGRAM c. könyvsorozat I. kötetében már olvashattunk információt a GENS/MONS párosról. Mindettől függetlenül sok-sok Olvasói levél tartalma azt bizonyította, hogy sokan vannak, akiknek ez a könyv nincs birtokában, s ha jól tudjuk ennek a kötetnek az LSI már az utánnyomását sem tervezi. Íme tehát elevenítsük fel, mit is tud ez a két jól ismert program.

GENS3 ASSEMBLER

Az általunk is tesztelt program felépítése:

by: GENS3 (8254 bytes)

Itt megjegyeznénk, hogy forgalomba került a GENS3 M ill. M2 kibővített változata is, amely hatékony microdrive ill. lemezes kezelést tesz lehetővé. Ez utóbbi főkérdésének hossza kb. 10.5 kbyte.

Az assembler LOAD "CODE xxxxx", vagy LOAD "GENS3"CODE xxxxx, utasítással tölthetjük be. Az xxxxx értéke tetszőleges, de célszerűen a RAM terület elején érdemes elhelyezni, mert mögé fog kerülni a forrásszöveg és a címtábla is.

LOAD "GENS3"CODE n esetén a program a RANDOMIZEUSR n utasítással indítható, az újraindításra két lehetőségünk van:

RANDOMIZEUSR (n + 56) - hidegindítást

RANDOMIZEUSR (n + 61) - melegindítást

A hidegindítás azt jelenti, hogy törődik a forrásszöveg és a címtábla is, míg melegindításnál csak a forráskódotól búcsúzunk, a szövegrész és a címtábla megmaradnak.

Indítás után az első kérdés a 'Buffer Size?', amelyre egy 0 és 9 közötti számmal válaszolhatunk. Ha csak 'ENTER'-t nyomunk, akkor beáll az alapértékre, ez 4. Itt állítjuk be a pufferek számát. Ezek a pufferek 64 byte-osak és a forrásszöveg beolvasásához is szükségesek lehetnek.

A GENS3 megszakítást nem engedélyez, így ha viszatérünk a BASIC-hez, akkor nekünk külön kell gondoskodni a megszakítás engedélyezéséről.

A 'TABLE SIZE?' kérdés a címtábla nagyságára kérdez rá (az 'A' nevű Editor-parancsnál). Elég itt is, ha csak 'ENTER'-rel válaszolunk.

Fordításnál különböző fordítási opciók közül választhatunk. Az 'option' kérdésre az alábbi opciókhoz tartozó számok összegét kell megadnunk:

- 1 - címtábla keletkezik a fordítás végén
- 2 - tárgykód nem keletkezik a fordítás végén
- 4 - nem kapunk assembly listát a fordítás végén
- 8 - az assembly listát nyomtatásra küldjük
- 16 - elhelyezi a tárgykódot
- 32 - kikapcsolja az ellenőrzést, a fordítás gyorsabb lesz
pl. a 63-os opció megadása jelenti a leggyorsabb fordítást.

A forrássorok sorszámot kapnak a BASIC-hez hasonlóan. A bináris konstansok %-kal kezdődnek (pl. %111 = 7). A címkék pedig L előtaggal szerepelnek (pl. L13). A címszámlálóra való hivatkozásnál \$ (dollárjel) használunk. A műveleteknél a modulus függvényre "?" jelre hivatkozunk, a logikai vagy függvényre

re "@" (kukac) jellel, míg a logikai kizáró vagy függvényre a "!" (felkiáltó)-jelet használjuk.

Assembler direktívák (fordítónak szóló üzenetek):

ORG n : a tárgykód kezdőcímét jelöli ki. Előtte csak komment sorok állhatnak. Az n lehet kifejezés is.

EQU : az EQU kifejezést címkének kell megegyeznie. Értékadó utasításoknak számitanak.

DEFB kifejezés, kif., kif... : a fentivel szemben mindegyik kifejezést 8 bite értékkel ki, és konstansként helyezi el.

DEFW kif., kif... : a fentivel szemben most 2 byte-on teszi le a kifejezést.

DEFS kif. : a kifejezésnek megfelelő számú byte-ot foglal le.

DEFN "string" : egy stringet tesz le a hosszának megfelelően (max. 256 byte).

kif. : ha a kif = 0 akkor a forrássorokat nem veszi figyelembe, amíg nem találkozik vagy az ELSE vagy az END utasításokkal.

ELSE : Ld. az if-nél.

END : ismét bekapcsolja a fordítót.

Megjegyzés : ezek a feltételes "ÁLKÓDOK" nem ágyazhatók be egymásba.

Az assembler parancsai:

Az assembler direktívákhoz hasonlóan nincs hatásuk (közvetlen hatásuk) a tárgykódra, csupán az assembly lista formáját módosítják. E parancsok a forrássor elején *-gal (csillaggal) kezdődnek, ezután betű (nagy betű) következik, és ezt esetleg "+" vagy "-" jelek követhetik. A sor hátralévő része tetszőleges lehet, kivéve az L+, L-, D+, D- parancsokat. A következő parancsok érthetőek el:

- *E (eject) : 3 üres sort emel a kérepyön vagy a nyomtatón
- *Hs (head) : *E valamint az "s" stringet kinyomatja
- *S : eddig a sorig listáz, a listázás bármely billentyűvel folytatható, viszont idő előtt leáll, ha *L vagy *S parancsokkal találkozik.
- *L- : e sortól kezdve már nem listáz
- *L+ : e sortól kezdve listáz
- *D- : a címszámlálót ezután HEX formában írja ki
- *D+ : a címszámlálót ezután DEC formában írja ki
- *C- : a sor után rövidített assembly listát ad
- *C+ : a sor után a teljes assembly sort kiírja
- *F (filename) : szalagról beolvass egy szövegfájl-t, amelyet lefordíthatunk, ennélfogva a szöveg csak minimális helyet foglal le a tárbán, amelyet kisebb adagokban olvashatunk be. A szövegfájl-t korábban a "T" parancsokkal vittuk ki a 'Buffer size?'-nek megfelelően. Ekkor a szövegfájl blokkokban helyezkedik el, és amikor az "F" parancsokkal beolvassunk egy blokkot, akkor azt rögtön le is fordítjuk. A blokkok beolvasása mindaddig tart, amíg be nem olvastuk valamennyi blokkot.

Megjegyzés : a "P" parancs erre nem jól! Tehát, ha a forrásszöveg nagyon hosszú, akkor először azt a "T" parancsokkal kivisszük, majd "F"-el visszaolvassuk kisebb adagokban.

Megjegyzés:

Az EDITOR parancsai (szövegfájl szerkesztéséhez)

A GENS3 indítás után közvetlenül az editor parancsait várja. A továbbiakban a Symbol Shift-et SS-sel, a Caps Shift-et CS-sel fogjuk rövidíteni.

Az editor a ">" jelet használja parancskéréshez.

l n,m : Sorokat beszúrhatunk az n. sortól kezdve m lépéssel. Kilépés CS + 1-re. A már létező sorszámú sorokat törli (felülírja). Egy szövegsor 64 karakter lehet (buffer size).

L n,m : Listáz az n-től az m-ig. Kilépés CS + 1-gyel.

K n : Listázáskor legfeljebb n sor látható.

D (n,m) : Töröl n-től m-ig. Egyetlen sort n=m révén törölünk.

M n,m	: Sorokat mozgathatunk át. Az n. sort az m-be viszi át. Az m elhagyható.
N (n,m)	: Átsorszámozza a sorokat. Az első sor sorszáma n, a lépésköz pedig m.
F n,m,l,s	: Egy f stringet keres az n és m sorok között. Ha ilyet talál, akkor leáll, és az 'E' (EDIT) paranccsal az f string az adott stringre cserélhető. A további keresés 'F'-el elintézhető.
E (n)	: Az n sorszámú sort a pufferba másolja át, ahol tetszés szerint módosítható, de szükség esetén ismét előhívható az eredeti sor is.

Az edit-nek több alparancsa is van, amelyet a CS-tel aktivizálhatunk. Pl. CS * * (Szököz) esetén meghagyja az eredeti karaktert, azaz megnoveli a soron belüli pointert, egészen a sor végéig.

CS0	: A Basic-hez hasonlóan karaktert töröl.
CS8	: A szövegpointert egy tab-pozícióval nagyobbra állítja.
ENTER	: A sor kész.
Q	: Kilép az edit-ből és törli a változtatásokat.
R	: Újra betölti az adott sort a pufferba, miközben a változtatásokat elfelejti.
L	: A sor hátralevő részét meghagyja és a pointert a sor elejére állítja.
K	: Törli az adott karaktert.
Z	: Törli az összes karaktert a pillanatnyi pozíciótól kezdve a sor végéig.
F	: A korábban megadott stringet keresi. Ha ilyet nem talál, akkor kilép az edit-módból.
S	: A korábban megadott stringet lecseréli a már megadott a stringre.
I	: Az adott pozíciótól kezdve karaktereket szűr be. Kilépés 'ENTER'-rel ezen belül a BASIC-hez hasonlóan működik a CS0 és a CS8-is.
X	: A sor végére állítja a pointert és a BASIC-hez hasonlóan szerkeszthető a sor.
O	: Felülírható az adott karakter. Kilépés 'ENTER'-rel, valamint a BASIC szerkesztő utasításai használhatók.

A magnetofon-kezeléssel kapcsolatos parancsok:

P n,m,a	: Szövegfile-t ment ki a szalagra az n-től kezdve m-ig folyamatosan, a néven.
G s	: Egy s file-t keres a szalagon és betölti azt. Ha már van a memóriában szövegfile, akkor e mögé tölti be a szalagról, miközben a teljes szöveget átsorszámozza.
T n,m,s	: A szövegfile-t blokkokban viszi ki az n sortól kezdve m-ig a néven. A blokk méretét a 'Buffer size?'-nél adtuk meg. A 'F' parancsnál olvassuk majd vissza a szövegfile-t.

Az editor parancsai fordításhoz és futtatáshoz:

A	: Lefordítja a szövegfile-t.
R	: Az ENT-nek megfelelő címtől futtatja a tárgykódot, amely RET-tel végződik.

Egyéb parancsok:

B	: Visszatérés a BASIC-hez.
C	: Lehetőség van a GENS1-ben írt programok konvertálására a GENS 3 alatt. Ekkor 'G'-vel beolvassuk a GENS1-ben írt programot, majd a 'C' paranccsal a GENS 3-nak megfelelő formára alakítjuk. Ettől kezdve már az ismert módon formálhatjuk a programot.
d	: A delimiter (határolójel) megváltoztatható a d string első karakterével.
p	: Kinyomtatja az alapértelmezéseket, ahol is a sorszámozók 1 - 32767 között lehetnek, míg a stringhosszak max. 20 karakteresek.
W (n,m)	: Kinyomtatja az n és m közé eső sorokat. A K értékének megfelelően bizonyos számú sor kinyomtatása után szünetet tart, amit tetszőleges billentyű lenyomása után folytat.

X	: Kinyomtatja a kezdő és a végcímet. Ekkor lehetőség van arra, hogy pl. MONS3 által létrehozott szövegfile-t a GENS3 editora átvegyen.
----------	--

A GENS3 által használt szövegfile-t az xxxxx + 54 és xxxxx + 55 végcímek határozzák meg, tehát ha egy szövegfile a RAM-ban van, majd betöltjük a GENS3-at és BASIC-ből kiadjuk a POKE xxxxx + 54, végcím (alsó byte) és POKE xxxxx + 55, végcím (felső byte) parancssort, akkor RANDOMIZE USR xxxxx nem jó, hanem RANDOMIZE USR xxxxx + 61 utasítással lesz a szövegfile a GENS3-nak megfelelő!

A GENS3 hibakódjai:

1. – a hiba ebben a sorban van
 2. – a mnemonikot nem ismeri fel
 3. – az utasítás rossz "formájú"
 4. – a szimbólumot többször is definiálták
 5. – a sorban illegális karakter van
 6. – az operandusok egyike illegális
 7. – a szimbólumként használt szó "foglat"
 8. – tiltott regiszterhasználat
 9. – sok a regiszter
 10. – a 8 bites kifejezés többre sikerült mint 8 bit
 11. – a JP (IX + n) és JP (IY + n) tiltott
 12. – az assembler direktíva formája hibás
 13. – illegális hivatkozás, pl. az EQU-ban olyan vál-tozó szerepel, amely még nem kapott értéket
 14. – osztás zérussal
 15. – szorzásnál fellépő túlcserdülés
- BAD ORG ! – hibás ORG cím fordult elő
 Out of Table space! – a szimbólumtábla számára kevés a hely
 Bad Memory! – nincs több hely a szövegfile részére

MONS3 MONITOR/DISASSEMBLER

A mi általunk tesztelt program hossza:

by: MONS3 (5760 bytes)

Mindenekelőtt töltsük be azt a programot, amit vizsgálni szeretnénk. Ezután töltsük be a MONS3 programot is:

LOAD=CODE xxxxx, ahol xxxxx tetszőleges, de a RAMTOP felé érdemes betölteni. A MONS3 indítása (feltételezett) első indításnál RANDOMIZE USR xxxxx, ismételt indításnál: RANDOMIZE USR (xxxxx + 29)
 A programból való kilépés az EDIT funkcióval (CAPS SHIFT + 1) történik.

A program indításakor a képernyőn legfelül egyetlen kiragadott disassembliált sor látható. A sor elején a memória-mutató (pointer) értéke látható, amelyre mindegyik parancs hivatkozik. Alatta a regiszterek és a regisztereknek megfelelő értéktől kezdve HEXADÉCIMALIS dump (memórialista) látható a regiszterekkel egy sorban. Az AF regiszterpár alatt 8 sorban a memória-mutató környezetének HEXADÉCIMALIS dumpja látható.

Itt is a továbbiakban a SYMBOL SHIFT jelet SS-sel, a CAPS SHIFT-et pedig CS-sel fogjuk rövidíteni.

SS + 4 (\$) : Az MP (memória-pointer)-nek megfelelő címtől egy lapnyi disassemblált listát ad. Visszatérés ismét a SS + 4-gyel, tovább bármelyik billentyűvel.

ENTER : MP = MP + 1

CS + 7 : MP = MP - 1

CS + 5 : MP = MP - 8

CS + 8 : MP = MP + 8

; (vessző) : MP = a nyíllal jelölt regiszter értéke.

G : Stringkereséshez. Ezután kétjegyű HEX számot adunk meg 'ENTER'-rel elválasztva. Kilépés szintén 'ENTER'-rel. Ismétlés kereséshez: 'N' (NEXT).

H : Decimálisból HEXADECIMALISBA konvertál.

I : (Int.copy) blokkmásoláshoz

J : Az adott (MP-nek megfelelő) címtől indít egy assembly programot. Ez a parancs végrehajtása előtt törölhető a CS + 5-tel.

SS + K : A program végrehajtása a PC (Program Counter) értékének megfelelően folytatódik (a legközelebbi BREAK-ig). Ennek akkor lesz szerepe, amikor egy programot lépésenként futtatunk (ld. SS + Z parancsot), de egy bizonyos pontig szeretnénk "azonnal" lefuttatni az assembly utasításokat (ezt a pontot BREAK-kel, 'W'-vel kellett beállítani), majd a ponttól kezdve ismét lépésenként futtatni a programot. Így a CALL utasításokat gyorsan kiadhatjuk.

L : Listáz egy blokkot az MP-nek megfelelően. Kilépés CS + 5-tel.

M : Az MP értékének tetszőleges beállítása.

N : Ld. a "G" parancsnál.

O : Relatív ugrás.

P : A memória egy részét egy megadott byte-tal tölti fel.

Q : Az EXX assembler utasításnak felel meg.

SS + T : BREAK-point-ot helyez el a jelenlegi utasítás mögé és a végrehajtást folytatja. Amennyiben a SS + Z érvényes, akkor a SS + 7 hatására pl. egy CALL utasítást azonnal végrehajtja, majd ismét "lépeget".

T : Disassemblál egy blokknyi memóriát és esetleg nyomtatóra küldi. Először a disassemblálendő szöveg kezdő és végcímét adjuk meg. Ezután a "Text:" kérdésnél vagy HEX címet adunk meg, vagy 'ENTER'-t ütünk. Ha egy címet

adunk, akkor ettől a címtől kezdve egy assembly forrásszöveget állít elő a disassemblálás alapján. Ezt a szövegfile-t adhatjuk át azután a GENS3 nak. E szövegfile előállítás után kilírhatja a végcímet is. E két cím birtokában beállíthatjuk a GENS3 ban a kezdőcímet az "X" alparanccsal, ill. ismerjük a végcímet is. Ekkor a "Workspace;"-re egy címet vár, amelyre egy "primitív" szimbólumtáblát tesz le. Elég 'ENTER'-rel válaszolni. Ha a "Text;"-re 'ENTER'-rel válaszolunk, akkor "csak" a képernyőn jelenik meg a disassemblált szöveg. Ezután ismét egy kezdő- és egy végcímet vár. Most kijelölhetünk a korábban kijelölt blokkon belül egy részblokkot, amelyet a MONS3 nem Z80-as utasításnak tekint, hanem konstansterületnek, és DEFB utasításként kezel. A "T"-ből CS + 5-tel lehet kilépni. Amennyiben az előállított szövegben olyan ugrás szerepel, amely a kijelölt blokkba mutat, akkor az abszolút cím elé "L" betűt helyez el. Amennyiben ismeretlen utasítással találkozunk, akkor egy NOP és csillagjel következik. Figyelem! Ha a forrásszövegben BREAK fordul elő (ld. "W"-t), akkor lehet hogy kettévágtuk az assemblerutasítást, és lehet, hogy a disassemblált szöveg "hamis".

U : Az "O" utasítással együtt létezhet. Amíg az "O" egy relatív ugrást hajt végre, addig az "U" visszatérést jelent. Ha több "O" is előfordul, akkor csak a legutoljára kiadott "O"-ra ugrik vissza, és a többi "O"-t nem veszi figyelembe.

V : Az "X" parancs fordítottjának tekinthető. Amíg az "X" egy abszolút ugrást hajt végre, addig a "V" visszatérést jelent. Csak a legutoljára kiadott "X"-nek megfelelően ugrik vissza, míg a többi "X" elveszik.

W : BREAK-point-ot helyez el a forrásszövegben. E címtől 3 byte-ot lecserél egy CALL utasításra.

X : Egy abszolút CALL vagy JP utasításnak megfelelő címre ugrik el.

Y : Az MP-nek megfelelő címtől ASCII karaktereket írhatunk be. Kilépés CS + 5-tel.

SS + Z : Egy lépést hajt végre az MP-től függően (lépésenkénti végrehajtásnál láthatók a regiszterek is).

SS + P : Nyomtatóra listáz.

QUARX • Your Sinclair

A pontszám-táblába írjuk be HI GREG, majd nyomjuk meg a <Z>, <S>, <D>, <G>, <H> és a billentyűket együtt. Ekkor kapunk 8 életet, és átkerülünk a következő szintre.

Animáció a képernyőn

Az animáció problémája már többször szerepelt a SpV. hasábjain is, ám a levélírók között igen sokan vannak, akik erről a témáról szívesen olvasnának hasznos információkat.

Arról már ejtettünk szót, hogy teljes képernyős animálás csak 6 db. képpel valósítható meg, 128K tulajdonosok ezt még 12-vel megtoldhatják. Sajnos azt hamar beláthatjuk, hogy a mozgás szaggatott, darabos. A gyorsítást gépi kódú segédrutin alkalmazásával érhetjük el, vagyis lehetővé kell tenni több kép eltárolásának, valamint gyors, egymás utáni előhívásának a lehetőségét. Erre a feladatra a **LASER BASIC** program önmagában is alkalmas, ám betöltése hosszadalmas, nem beszélve a memóriacsökkenés tényéről.

A most általunk ismertetésre kerülő program eleget tesz a fontos kritériumoknak, használata, illetve használatának elsajátítása pedig egyszerű.

Kezdjük mindjárt a nehezével, valamely, nekünk szimpatikus editor/assembler programba vigyük be az alábbi kódot:

```

10          ORG 50000
20 CIM     LD A,(NOP1)
30        CP 235
40        JR NZ,JO
50        LD HL,VIZSG
60        DEC (HL)
70 ODA     LD A,(NOP1)
80        XOR 235
90        LD (NOP1),A
100       LD (NOP2),A
110       LD A,0
120       LD (SZAM),A
130       JR JO
140 TOLTO  LD A,(NOP1)
150       CP 0
160       JR NZ,AT2
170       LD HL,16384
180       LD (KEPER),HL
190       LD A,1
200       LD (VIZSG),A
210       JR ODA
220 AT2   LD HL,VIZSG
230       INC (HL)
240 JO     DEFB 1
250 VIZ    DEFB 5
260       DEFB 0
270       DEFB 17
280 FUGG  DEFB 40
290       DEFB 0
300       CALL SZORZ
310       DEFB 1
320 SZAM  DEFB 1
330       DEFB 0
340       CALL SZORZ
350       LD HL,END
360       ADD HL,DE
370       DEFB 17
380 KEPER DEFW 16384
390       LD A,(FUGG)
400       LD B,A
410 C1    PUSH BC
420       PUSH DE
430       LD A,(VIZ)

```

```

440       LD C,A
450       LD B,0
460 NOP1  NOP
470       LDIR
480 NOP2  NOP
490       POP DE
500       INC D
510       LD A,D
520       AND 7
530       JR NZ,C2
540       LD A,E
550       SUB 224
560       LD E,A
570       JR NC,C2
580       LD A,D
590       SUB B
600       LD O,A
610 C2    POP BC
620       DJNZ C1
630       LD A,(SZAM)
640       DEFB 254
650 VIZSG DEFB 1
660       JR C,VG
670       LD A,255
680 VG    INC A
690       LD (SZAM),A
700       RET
710 SZORZ LD HL,0
720       LD A,16
730 LOOP  ADD HL,HL
740       RL C
750       RL B
760       JR NC,SKIP
770       ADD HLL,DE
780 SKIP  DEC A
790       JR NZ,LOOP
800       EX DE,HL
810       RET
820 END   END

```

A programot fordítsuk be az egyszerűség kedvéért 50000.-re, majd mentjük ki. Ismételt betöltés után a program már futásképes (ne felejtjük el betöltés előtt kiadni a **CLEAR 49999** utasítást!).

A program használata

Az eltárolandó képek letapogatása mindig a képernyő bal felső sarkából indul. A tárolandó ablak méretét egyszerűen rögzíthetjük: **POKE 50054,x: POKE 50057,y**, ahol x az ablak vízszintes, míg y a függőleges mérete képpontokban. Az ablak tartalma a **RANDOMIZE USR 50029** utasításra mentődik el a memóriába, ugyanakkor a képek száma emelkedik 1-gyel. Ha újra meghívjuk a rutint, újabb kép mentődik el, és így tovább. Ha az összes képünk a memóriába került, próbáljuk meg előhívni ezeket: **RANDOMIZE USR 50000**, és az első eltárolt képünk megjelenik a bal felső sarokban. Ha újra kiadjuk a **RANDOMIZE USR 50000** utasítást, megjelenik a következő is, s a többi szintén, sorban. A képet a képernyőn pozícionálhatjuk is: az **50073/50074** címen. Ha a **POKE 50073,10: POKE 50074,72 (18442)** utasítások után hívjuk meg a **RANDOMIZE USR 50000** parancsot, úgy a kép a képernyő közepén fog megjelenni. Akár egy megadott számú képet is előhívha-

tunk: POKE 50063,képszám, majd RANDOMIZE USR 50000, és a kiválasztott kép kerül a képernyőre.

Gyorsan a közepébe vágunk, úgyhogy nem árt, ha az eddigieket egy kis mintapéldán keresztül bemutatjuk:

```
10 FOR F=0 TO 19: PLOT 20+F,15
5+f: LET a=2*f: DRAW -a,0: DRAW
0,-a: DRAW a,0: DRAW 0,a: RANDOM
IZE USR 50029: CLS: NEXT f
20 FOR F=18 TO 1 STEP -1: PLOT
20+F,155+f: LET a=2*f: DRAW -a,
0: DRAW 0,-a: DRAW a,0: DRAW 0,a
: RANDOMIZE USR 50029: CLS: NEXT
f
30 PAUSE 3: RANDOMIZE USR 5000
0: GO TO 30
40 FOR F=0 TO 19: CIRCLE 20,15
5,f: RANDOMIZE USR 50029: CLS: N
EXT f
50 FOR F=18 TO 1 STEP -1: CIRC
LE 20,155,f: RANDOMIZE USR 50029
: CLS: NEXT f
60 PAUSE 3: RANDOMIZE USR 5000
0: GO TO 60
```

Adjuk ki: RUN 10, azután BREAK, majd RUN 40. Persze a BREAK-et nem azonnal kell megnyomni, várjunk egy kicsit!

Láthatjuk, hogy a program használata milyen egyszerű. Alaphelyzetben egy 40x40-es ablakot definiáltunk (5x40, azaz 5 karakter = 40 képpont). A kimentést az 50000. címtől kell kezdenünk. A kód 143 byte hosszú, + a képek (vízsz.*függ.*(db.+1)). Próbáljuk most elhagyni a 30. sort, és RUN.

Végezetül annyit, hogy amennyiben az ORG változik, akkor assemblálás után érdemes rögzíteni az alábbi címeket:

TOLTO	- a képek betöltése
CIM	- a képek kirakása
SZAM	- a kirakandó kép száma
VIZ	- szélesség karakterben
FUGG	- mélység képpontban
KEPER	- a kép bal felső koordinátája két byte-on
VIZSG	- a memóriában lévő képek száma
END	- ettől a címtől kezdődnek a képek a memóriában.

Zene 3 csatornán

Az LSI ATSz gondozásában megjelent SINCRAIR SPECTRUM JÁTÉK ÉS PROGRAM c. könyvsorozat 5. kötetében ötletkelet kaphatunk egy 3 csatornás hanggenerátor megépítéséhez, programozásához. A SpV 128K rovatának hasábjain több részben leközelt "A hanggenerátor programozása" c. információk sok hasonlóságot mutattak ahhoz, ami a könyvben megjelent, egy szempontú eltérés azonban már az elején kiálló hatott a témával foglalkozó Spectrumos elérés mutatkozott a port címek árakéiban. Ez valóban azt jelenti, hogy a könyvben ismertetett hanggenerátor, és a 128K gép hanggenerátora kompatibilis egymással, sőt a Dk'ronics gyártmányú hanggenerátorral is. Hogy tudnánk egyszerűen megoldani, hogy a 48K géppel rendelkezők is élvezhessék a 3 csatornás dallamok örömeit? Nos a hardware átírás, vagyis egy port-kapcsoló kiegészítő-kör-rendszer beépítése a hanggenerátorba bonyolultabb megoldás lenne, mint magának a programnak az átírása. Ehhez csak annyi dologunk van, hogy megkeressük a programban a zene generálásért felelős rutint, és abban a módosításokat megtalálva elvégezzük. Mint tudjuk, a 128K 3 csatornás zenei portjai a/b = 65533/49149. Ez az érték a mi általunk közölt interface esetén, a/b = 159/223, míg a Dk'ronics típusú interface esetén a/b = 63/95.

Most megoldást kínálunk néhány program esetére, célzerű a módosítást MULTIFACE-szel elvégezni, majd a módosított programot kimentve, a hanggenerátor interfacéit a gépbe dugva, a a programot visszatöltve élvezhetjük is munkánk eredményét:

```
A.T.F.: POKE 32484,a POKE 32485,b POKE 32488,14 POKE 32489,b
Bobsleigh: POKE 34489,a POKE 34490,b POKE 34493,14 POKE 34494,b
Cybernoid: POKE 61361,a POKE 61362,b POKE 61365,b POKE 61367,0
FOR I=35889 TO 25885: POKE I,0: NEXT I
Exotik: POKE 48010,a POKE 48011,b POKE 48014,14 POKE 48015,b
```

A módosításokban a/b értéke a lent leírtaknak megfelelő. Reméljük, hogy a programozásban jártasabb Olvasóink nekikéntnek átírítani néhány – erre alkalmas – programot.

Orbiter

Az örökélet bevitelét a BASIC(286)/3270 file térképpel rendelkező verzió mutatjuk be.

A játék alatt a program BREAK-elhető. Az 5. sort írjuk át:

```
5 IF L=3 THEN GO TO 2, majd
```

```
RUN (ENTER)
```

és a játék örökéletet tesz.

Hol vagy CoVboy?

Kedves SpV-sek!

Köszönöm az ajándékazétját, amit a THE WORLD OF BARBARIAN leírásért küldtetek. Szép ajándék volt a kezdő COMMODORE-os unokatesómnak. Kár azért a két programért, ami így jelentkezett be: LOAD ERROR, majd lefagyott. (Az unokatesódnak is tudjuk ajánlani a CoV 4. számában megjelent mesét! SpV) Mindez ellenére sikerült vele nagy örömet okoznom (és különben is ajándék lónak ne nézd a fogat. Nem igaz?)

Azért írtam újra, hogy dőcsérjem a lapot főleg az újabb számok (a CoV-ot is beleértve) alapján.

A POKE-okat egyre jobban választják meg, és míg a régebbi számokban 8-ból csak egyet volt érdemes használni, újabbak már csak a fejemet kapkodom, hogy hol is kezdjem. Aztán itt van CoVboy, aki senkit sem kímél, és lehet, hogy sokak kedvét elveszi majd a levélírástól. Én is csak azért vettem tollat a kezembe, mert abban bírok, hogy az SpV-ből ez a rész kimarad, hiszen CoVboy még elmegy, de hogy nevezzenek el Spectrumosan? (Ez persze csak vicc, én lennék a legholdogabb, ha az SpV-k utolsó oldalain is egy CoVboy-féleség poénkodna).

—Egyébként kíváncsi lennék, mennyi kelt el az SpV-ből (a 20-ból) (Lehet, hogy furcsa, de amikor ezeket a sorokat írok, még mi is kíváncsián várjuk ezt az információt. SpV) Az is érdekelné, megjelenik-e a Spectrum Játék & Program 6.7 Mit tudok róla? (Amikor ezt a számot kézbeveszed, ez utóbbi kérdésre már megkaptad a választ! SpV)

Üdvözlök mindenkit (a szakállast és a CoVboyt is) (Átadtuk! SpV)

Ü: Sokat gondolkodtam CoVboy elnevezésén (a törlecke bánta), de ennél jobb nem jutott eszembe: ZÉIKSZ.

IHIDI OLIVÉR, SZOMBATHELY

(SpV: A CoVboy-hoz hasonló néven mi is gondolkodtunk, de sajnos annyira frapriánsat nem tudunk kiötlölni, így hát maradtunk a jól bevált SpV megnevezésnél.)

Hol az emblémám?

Tisztelt SpV!!!!

A múltkoriban küldtem Önöknek egy levelet, melyben egy SUPER-HYPER emblémát ajánlottam fel a SpV-nek, valamint bátorkodtam megkérdezni, hogy szükségük van-e a Navy Moves II. teljes leírására és térképére + indítókód (s ha igen, mennyiért?) 3 hét után jött a válasz: "TC. Kérdésére megtalálhatja a választ a SpV következő számaiban". Hát ezt kissé kétféle, szóval vagy elkavartott a levelem valahol, azaz összecserélték a választ, vagy újabb minden levélre ezt a "kimerítő" választ adják, olvashatatlanul, ami utóbbi kissé — elnézést — bunkóság. Tehát: jó lenne, ha előkértenék előző leveleimet (az emblémával együtt) és válaszolnának is rá normálisan (s ne 3 hét múlva).

Ü: Azért harag nincs, ugye?

PUIZ ANDRÁS, BUDAPEST

(SpV: Harag az természetesen nincs. Az embléma ügyében már szóltunk néhány

keresetlen szót az előző szárn levelezésében. Akkor elmondtuk, hogy a következő számban meg fogjuk mutatni az Olvasóknak. Most ott virít a hátsó borítón. Ennél előbb nem sikerült volna ezt megtennünk, hiszen egy-egy kép színbontása elérheti akár a 30 napot is, plusz érdemes még beszámítani az újság nyomdai áttutását is. Nem minden esetben szoktuk a fent említett "kimerítő" választ küldeni levelezőinknek, ám ezen levelezési rovat éppen azt a célt szolgálja, hogy a közérdekűnek talált levelekre itt reagáljunk. Sok értelme nem lenne ugyanezt borítékba téve elküldeni, éppen elég az ide be nem főző leveleken átrágnunk magunkat. A 3 hétről pedig annyit: levelekre akkor tudunk válaszolni, amikor az időnk is enged. Ezt pedig az befolyásolja, hogy elsődlegesen munkahelyünkön kell helytállnunk, meg kell írni, és el kell küldeni az újságot, s nem utolsó sorban még egy kis magánéletnek is bele kell férni a dologba. A Navy Moves ajánlatot köszönjük, egyébként nincs rá szükségünk. SpV)

Kérdések, vélemények...

T.Szerkesztőség!

A Spectrum Világ 20.számában közölt keresztrejtvény a meglepőtelenség határát súrolta. Egy keresztrejtvény miatt nem érdemes, sem latin szótárt, sem orvosi szótárt vásárolni. Sokkal nagyobb baj, hogy a földrajzi neveket sem lehet megtalálni még a világtalpasban sem. Említhetnék a rejtvényben szereplő NSZK folyót, valamint a Bantry-öböl több ezer kilométerre van a rejtvény meghatározásban szereplő Fokvárostól is). De ezek tulajdonképpen csak kellemetlenségek.

Nagyszerű ötlet volt a dobszimulátor program közlése. Nekem ugyan csak 48K-s gépem van, de hozzákapcsoltam a hanggenerátort és a programot némi átalakítás után én is ki tudtam próbálni. Sajnos nem vagyok biztos benne, hogy hibátlanul írtam-e át, mindenesetre működik. Ha lehet, egy rövid cikkben térjenek ki arra, hogy egy programban hogyan lehet megtalálni a zenei részeket. A Chronos c. programnak (és persze még néhánynak) nagyon jó a zenéje, pedig csak a BEEP-et használja (persze ebben nem vagyok biztos, mindenesetre 48K-s gépre íródott). Találkoztam olyan programmal is, amely a menüben a SOUND opcióban megkérdezi, hogy 48K, 128K-s a gép tehát a 128-as hanggenerátort is használja. Arra gondoltam, ha a gépemre rácsatlakoztatom a hanggenerátort (AY-3-8912-vel) és átírom a csatorna címeiket, a 128-as opciót választva nekem is "szépen fog dalolni" a gépem? Vagy ez nem ilyen egyszerű mert 128-as ROM rutinokat is meghívhat közben? Az alábbi helyettesítésre gondoltam:

Bredeti	Átírt
LD BC,65533	NOP
LD A,7	NOP
OUT (C),A	LD A,7
LD BC,49149	OUT (159),A
LD A,56	NOP
OUT (C),A	NOP
RET	LD A,56
	OUT (223),A
	RET

Mivel a külső hanggenerátor címzéséhez két byte-tal kevesebbre van szükség, a felszabaduló byte-ok helyettesíthetők-e 2 db. NOP-pal? (Igen! SpV) Ha igen, az nagyon jó, mert nem kell átírni a relatív ugrásokat sem. Lehet, hogy meglehetősen naív az elképzelésem és nem fog működni, mindenesetre megpróbálom.

A programozástechnika sorozatban az örökélet és végtelen energia megszüntetéséről is írjanak, mert az legálább olyan bosszantó, ha 255 élete van a játékosnak, mintha csak a pillanatok alatt elfogyó 3 élet áll rendelkezésre.

Ne csüggedjenek, nem minden Spectrumos pártolt el Önöktől, bár azt hiszem, ez gyenge vigasz, ha a lap csak annyira fog, hogy az már ráfizetéshez vezet.

HIRTH TIBOR, BACSALMAS

(SpV: Rejtvény ügyben röviden: A rejtvény egy ismerőcsünk készítette. Az igaz, hogy közgazdasági és nem földrajzi végzettséggel rendelkezik, de nem hivatásos rejtvényt készítő, és nem is voltak sosem ilyen ambíciói. Az esetleges nehézségekről az Ő nevében is elnézést kérünk, a továbbiakban nem kell azírni, mert a rejtvény iránti érdeklődés zuhanó tendenciája, valamint a beérkező megfejtések egyre csökkenő száma miatt ezen az oldalon inkább ezt a rovatot helyezzük el.

A hanggenerátor ügyben annyit, hogy néhány szót ez ügyben szólnunk itt balra a másik oldalon, tekintve a programok különbözőségét, erre a dologra egységes receptelt nem tudunk szolgáltatni, további sikeres próbálkozásait kívánunk!

Az örökélet megszüntetésének kérdése nehéz dió, hiszen megtalálni, és kiírni az életet csökkentéséért felelős részt valamilyen könnyebb, mint annak a fordítottját művelni. Sokszor elég nehéz lenne pl. egy 20-30 byte hosszán kinullázott memóriaterületet visszairni. Erre inkább nem vállalkozunk. SpV)

Hanggenerátor kerestetik!

Kedves SpV!

Olvastam a "Sinclair Spectrum Játék és Program 5"-ben hogy Rucz Lajos készített egy 3 csatornás stereo hanggenerátort. (Ezt hol olvastad? SpV) Mivel én 128K játékokat szeretnék írni, nagy szükségem lenne rá nekem is, ezért szeretnék egyet vásárolni, ezért szeretném kérdezni, hogy Rucz Lajos eladásra is készíti-e, és ha igen, mennyibe kerül? (Először is: Rucz Lajos lyet nem gyárt, főként nem azérlában. A leközölt adatok alapján a megépítés — természetesen megfelelő hozzáértéssel — nem lehet probléma. Másodszor: Ha 128K géped van, erre nem nagyon lesz szükséged, ha pedig 48K-s géppel rendelkezel, úgy nem tanácsoljuk, hogy 128K-s programokat kezdj el rajta fejleszteni, a port címek ugyanis nem stimmelnek, bizonyára már észrevetted, hogy éppen erről volt szó az előző levélben, és erről papírtunk az előző oldal végén is! SpV)

Egy trükköt most is leírok (magnó trükköt), ami előző levelemből kimaradt, és egy segítséget a spanyol programok felhasználóinak.

1. A Spectrum-jack "EAR" aljzatáról rakjuk át a kábelt a "MIC"-be. Töltésnek be egy jétkot, és a töltésnek nincs hangja! (Ezt inkább nem kommentáljuk! SpV)

2. Sokszor gondot jelenthet spanyol programok irányító gombjainak átdefinálása, ha nem tudunk spanyolul, vagy ha nincs kezünkben spanyol-magyar szótár. Ezért leírtam az előforduló irányokat, stb.

Derecha - jobbra, Izquierda - balra, Arriba, Subir - fel, Abajo, Bajar - le, Fuego, Disparo, Cambio - tűz, Frenar - lassítás, Acelerar - gyorsítás

BÁTI ANDRÁS, BUDAPEST

(SpV: Ez utóbbi segítséget köszönjük!)

Nyák nincs?

Tisztelt Cim!

Spectrum Világ 19. számában közölt interface nyákot szeretnék vásárolni 1 db-ot, amennyiben lehetséges.

ZALAVÁRI MIKLÓS, GYŐR

(SpV: Nyákok sokszorosításával, ill. forgalmazásával mi nem foglalkozunk. Talán egy klubot vagy egy hardware-es ismerőst kellene megkeresni!)

(At)váltás

Megszerezném tudni, hogy hogyan lehet átváltani 48K-os gépen BASIC-ból gépi kódra, ugyanis az eddigi Spectrum Világ részeitől nem tudtam meg.

VÉKONY GÉZA, BUDAPEST

(SpV: Ez aztán a szegénységi bizonyítvány. Már mint nekünk, hogy ezt eddig nem írtuk le, pedig milyen hosszú is volt ez a Gépi kód tanfolyam. Ha nincs a gépeden M/C billentyű, úgy lehet, hogy a géped már ISSUE 9956, azaz automatikusan meg tudja különböztetni, hogy BASIC, vagy gépi kódú programot futtasz!)

Néhány észrevétel...

Tisztelt Szerkesztőség!

Néhány észrevételt szeretnék közölni. Nem tudom, feltűnt-e már Önöknek a Tarzan c. játék térképén található hiba. Ha még nem, akkor elárulom, hogy hiányos. A térkép-melléklet Tarzan feliratainak 'AR' betűi felett található pályán a lejárati hiánya miatt egy terület rész hiányzik (lehet, hogy máshol is?). A játék értelmére azonban még nem tudtam rájönni. Az ismerőseim sem. Szerintem egy halom tárgy felvétele után vala-

miféle cselekvést kell elvégezni. Ha mégsem, kérem írják meg a könyvsorozat következő részében, hogy mégis mi a cél.

A másik kedvencem a The Planets Two című program. Sokáig idegesített a 'LOGON' ismeretlenség hiánya, míg aztán egyszer belemásztam. Íme az eredmény: LOGON = MARTECH. Ezután bejelentkezik az adatbázis. A parancsok: HELLO, HELP, COMET, METEOR, ASTEROID, LASER, ACCESS, GAMES, PLANET, CODE, MERCURY, VENUS, EARTH, MARS, JUPITER, SATURN, URANUS, NEPTUNE, PLUTO. Az ACCESS=xxxxxx formában meg lehet adni a magasabb szint kódját, ahol xxxxxx=5012753. Ezután már élnek a következő parancsok: LASERON, LASER-OFF. Az adatbázisból a LOGOFF parancsral léphetünk ki. Viszont itt sem ismerem a célt (pedig biztosan érdekes lehet).

Végül néhány POKE: Pulse Warrior: POKE 57791,0 - Cybernoid Two: POKE 36198,0 - Afterburner: POKE 37030,182 POKE 37935,182 - R-Type: POKE 37525,0 (12 valószínű csak MIKRO-POKER-rel működik). Üdvözzel!

KAJTAR ZSOLT, BONYHÁD

(SpV: A térképkiegészítést itt alul közöljük! A többi stuffért köszli)

Tartalomjegyzék

1	<i>Az Olvasó a szó</i>	1
2	<i>Játékismertető</i>	4
	- THE GAMES: SUMMER EDITION	4
	- XYBOTS	4
	- THE UNTOUCHABLES	5
	- OPERATION THUNDERBOLT	5
2.1	<i>The Bard's Tale (Electronic Arts)</i>	6
2.2	<i>Heavy On the Magick! (Gargyle Games)</i>	11
3	<i>ENTERFACE (Enterprise melléklet)</i>	15
4	<i>Ismeretlen nyelvek (Micro-PROLOG: Gépi kódú bővítés)</i>	19
5	<i>Ismeretlen nyelvek (HISOFT 'C' Compiler)</i>	22
6	<i>Advanced Header Reader (LERM)</i>	25
7	<i>GENS3/MONS3 (HISOFT)</i>	26
8	<i>Programozástechnika</i>	29

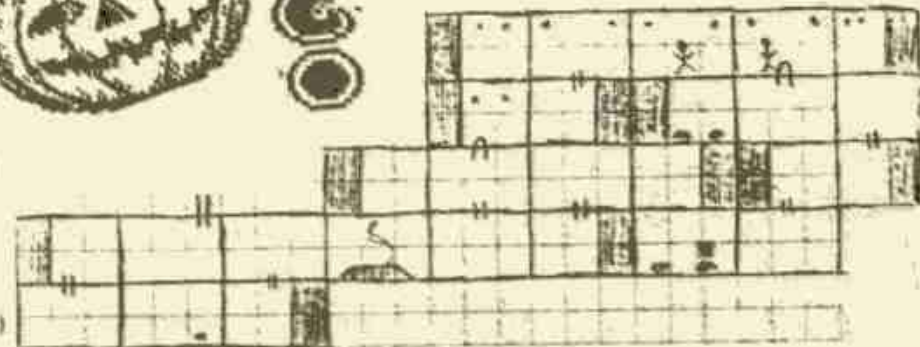
SpV. 13.rész, 10.oldal

Psi Chess

- Pause Option
P + 1: eredeti tábla;
P + 2: királyok és bástyák;
P + 3: királyok
- Game Option (Change)
J + 1: kurzor irányítás
vezérlés: 1,2,3,4,5 billentyűkkel
Kilépés: K
- M + 3 (Change)
Z + 1,2,.....,7,8
Bill. nyomás, lépés „zaja”
(1 - kikapcsol; 2-8 - hangerő)



TARZAN



A MŰSZAKI KÖNYVKIADÓ KANDÓ KÁLMÁN KÖNYVESBOLT

(Budapest, V. Bajcsy-Zsilinszky út 20.)
ajánlja a számítástechnikai könyvek iránt érdeklődő
olvasóknak, az újonnan megjelenő műveket:

BOISGONTIER, J. - DONAY C.

TURBO PASCAL FÁjlKEZELŐ ALKALMAZÁSOK

A Turbo-Pascalt a fájlkezelési lehetősége teszi megfelelő eszközzé az ipari és gazdasági feladatok megoldásához. A könyv első része röviden összefoglalja a programozási nyelvet, a második, hosszabb rész pedig számos példát bemutatva fájlkezeléssel foglalkozik. A fájlkezelés megértése jó kiindulási alapot teremt az adatbázisok használatához.

Kb. 160 oldal, 249,- Ft

KÖRTVÉLYESI GÉZÁNÉ

TURBO PASCAL 4.0 (lapozgató sorozat)

A szerző megadja az utasítások, eljárások, függvények pontos szintaxisát, felhívja a figyelmet a buktatókra, és példákat közöl. A felhasználó nem kézikönyvet kap, hanem olyan segédeszközt, amelynek minden számítógép mellett ott kell lennie. Az információ gyors megtalálását a funkció, ill. ábácé szerinti csoportosítás segíti.

Kb. 200 oldal, 180,- Ft

KÖRTVÉLYESI GÉZÁNÉ

TURBO PASCAL 4.0 GRAFIKA (lapozgató sorozat)

Az előző könyv folytatása, amelyben a grafikus utasítások szerepelnek.

Kb. 150 oldal, 180,- Ft

MAGYARI ISTVÁN

OPEN ACCESS (lapozgató sorozat)

Az Open Access integrált szoftver a kevés számítástechnikai gyakorlattal rendelkező felhasználói kör számára olyan eszköz, amely széles körű szolgáltatásaival (adatbázis-kezelés, számolótábla, szövegfeldolgozás, kalendárium, grafika, kommunikáció) képes az alkalmazói igények kielégítésére, kezelése pedig gyorsan és egyszerűen elsajátítható. E kötetben az alapvető tudnivalók összefoglalása után tömören, a formai előírások pontos betartásával ismerteti a programmodulok működését és a parancskészlet használatát. Külön fejezet foglalkozik a hibaüzenetekkel.

Kb. 200 oldal, 180,- Ft

ORBÁN KATALIN

PROFESSIONAL COBOL (lapozgató sorozat)

A Lapozgató sorozat e kötetének témája a PROFESSIONAL COBOL nyelv és rendszer, amellyel megszerkeszthetjük, szintaktikusan elemezhetjük és végre is hajthatjuk programunkat. Menürendszerével hibafelderítést, tesztelést végezhetünk. A könyv ismerteti az utasításokat, összefoglalja a direktívákat és a hibaüzeneteket.

Kb. 200 oldal, 180,- Ft



Budapest XIII., Sallai u. 28.

személyi számítógépek
és tartozékaik javítása

SINCLAIR SPECTRUM
javítás 48 óra alatt

6 hónap garancia

NYITVA: hétfőtől - péntekig

9.30-18 óráig,

szombaton: zárva

EGYSZERŰSÍTETT RENDELÉS A RÉGEBBI SZÁMOKRA

A SPECTRUM- és COMMODORE VILÁG eddig megjelent számaikat utánvétellel lehetett megrendelni. A postai költségek (nem szabványméret, súly, utánvételi díj, ajánlási díj) mértéke már 1989-ben is sokakat elriasztott attól, hogy utánvétellel rendeljen tőlünk régebbi számokat. 1 számra 33,- Ft, 2-5 számra 38,- vagy 43,- Ft – függően a súlytól – postaköltséget kellett fizetni. Az 1990. január 8.-i postai áremelkedések ezt az értéket tovább növelték. Belépett a levélforgalomban a helyi és a távolsági díjszabás, ebből adódóan az utánvétel díjtételei a következőképpen módosultak:

	budapesti megrendelő	vidéki megrendelő
1 szám	44,- Ft	49,- Ft
2-5 szám	50,- vagy 56,- Ft	55,- vagy 63,- Ft

(attól függően, hogy a küldemény súlya mikor lép át a súlyhatáron)

Ez azért érdekes, mert pl. a SpV. 9. száma majdnem csak fele olyan súlyú, mint a 14.szám. 5-nél többet pedig egy borítékban nem tudunk küldeni, mert csak csomagként lehetne feladni, ez pedig újabb bonyodalmakat okozna. Ezek a tények sajnos sok Olvasót visszariasztottak attól, hogy rendszeresen rendeljenek tőlünk a régebbi számokból, hiszen 1 szám megrendelése esetén mintegy a dupláját kell fizetni a kiadvány árának. A régebbi számok megrendelését egyszerűsíteni szeretnénk a következők szerint:

- a megrendelés magánszemélyek esetében hagyományos rózsaszín postai utalványon történik
- a megrendelés közületek esetében MNB átutalással ill. levél útján történő megrendeléssel történik
- a csekken feladott megrendeléseket a következő címre és számlaszámra kérjük:

SPECTRUM VILÁG

Fk: Rucz Lajos

OTP körzeti fiók Bp.XI.Irinyi u.30. 1117

MNB 218-98426/31624-4

- A magánszemélyeket kérjük, hogy megrendelésüket CSAK A CSEKKEL jelezzék (egy párhuzamosan elküldött levél adminisztratív bonyolítja az ügyintézkést) ! A rózsaszín belföldi postautalvány középső szelvényének hátoldalára – a közlemény rovatba – kérjük megjelölni a kért kiadvány(ok) sorsszámát (pl. 'SpV 1,2,8,15 és CoV 1 megrendelése'), és a számlaszámot (MNB 218-98426/31624-4) is.
- Közületek a megrendelő levelükkel párhuzamosan a kiadványok ellenértékét és a postai díjat MNB átutalással teljesítsék. A kiadványokat az átutalás teljesítése után postázzuk.
- Az ilyen formájú átutalás banki átfutása kb. 3 hét, erről nem mi tehetünk.
- A megrendelés díjtételei:

A SPECTRUM VILÁG 1-13. számai 39,- Ft-ba kerülnek, a 14. számtól 49,- Ft a fogyasztói ár. A COMMODORE VILÁG c. kiadványok fogyasztói ára ugyancsak 49,- Ft. A hagyományos levélforgalom díjtételei is jelentősen növekedtek, ezért az ilyen megrendeléseket nem levélként, hanem hírlapfeladás formájában teljesítjük, jelenleg ez a legkedvezményesebb postázási forma. Egy borítékban technikailag továbbra is csak 5 újság postázását tudjuk megoldani. 1, ill. 2 újság postai díja budapesti megrendelőinknek 8,- Ft, vidéki megrendelőinknek 12,- Ft. 3-5 újság postai díja budapesti megrendelőinknek 12,- Ft, vidéki megrendelőinknek pedig 18,- Ft. Ez így lényegesen kedvezőbb mint az eddigi utánvételes rendszer. Igaz, a pénz átutalásával járó költségek ez esetben a megrendelőt terhelik. Nézzünk a megrendelésre két példát. Egy budapesti Olvasónk szeretné megrendelni a Spectrum Világ 3. és a Commodore Világ 1. számát. Ez esetben $39 + 49 + 8 = 96,-$ Ft-ot kell részünkre átutalnia. Egy vidéki megrendelőnk kíváncsi a Spectrum Világ 7-18. számaira, ez esetben $39 + 39 + 39 + 39 + 39 + 18 + 39 + 39 + 49 + 49 + 49 + 18 + 49 + 49 + 12 = 550,-$ Ft-ot kell számunkra elküldenie (ekkor 5 számra 18,- ill. 2 számra 12,- Ft a postaköltség).

Az eddigi információkkal ellentétben továbbra is elfogadjuk azt, ha valaki utánvétellel rendel a régebbi számokból. Az Olvasóra bízunk, hogy melyik megrendelési formát választja. Az utánvételes valamivel drágább, viszont gyorsabb, a mi általunk javasolt olcsóbb, de a banki átfutás miatt kb. 3-4 hét átfutásra mindenképpen számítani kell!